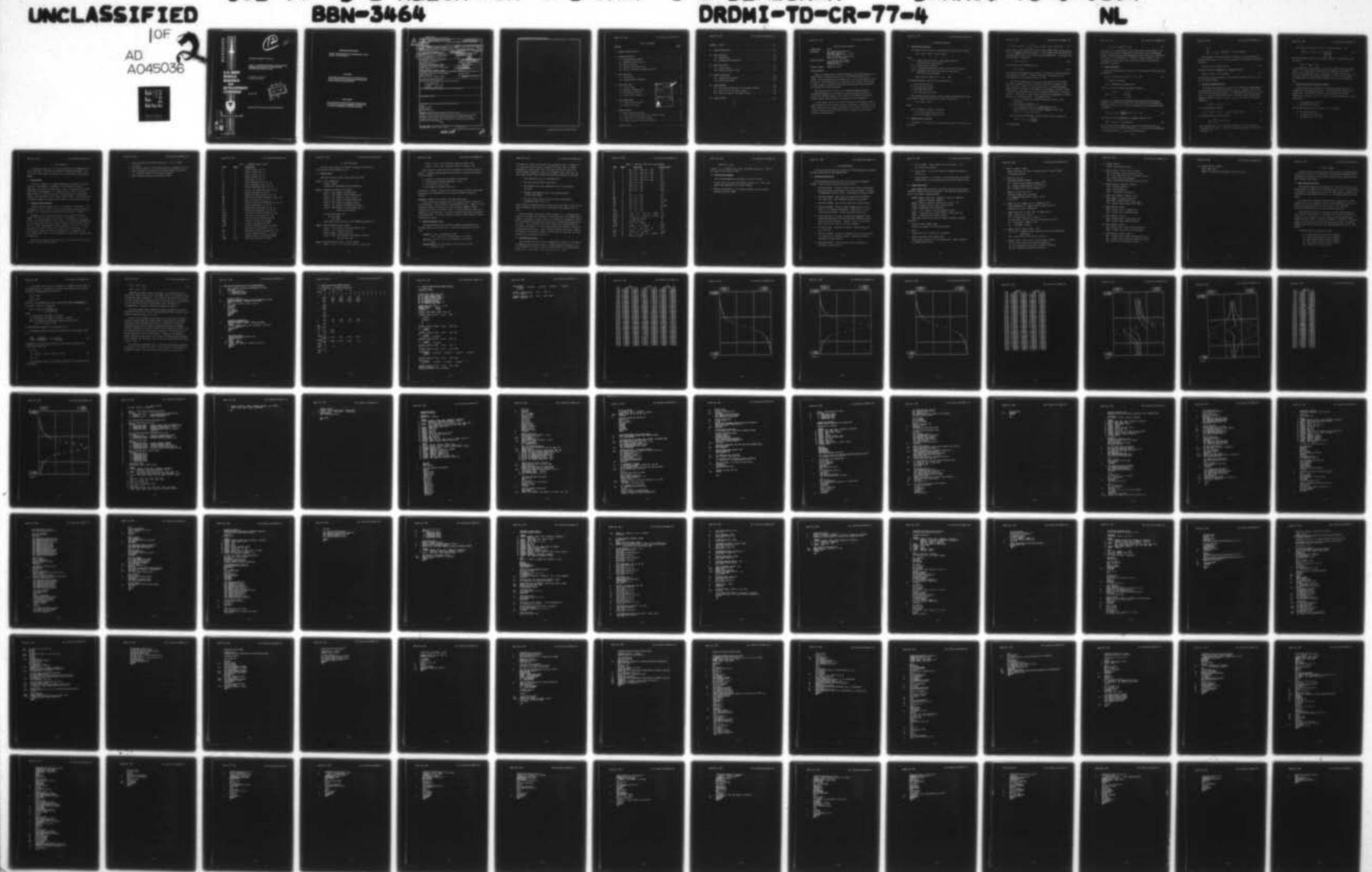
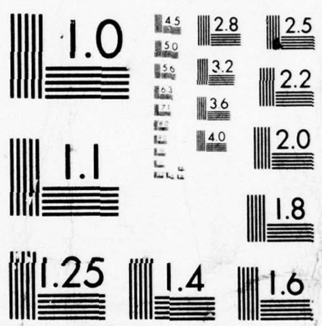
AD-A045 036 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS F/G 9/2
TIVAR: A COMPUTER PROGRAM FOR PREDICTING ENSEMBLE STATISTICS IN--ETC(U)
JUL 77 D L KLEINMAN, S BARON, J E BERLINER DAAH01-76-C-0194
UNCLASSIFIED BBN-3464 DRDMI-TD-CR-77-4 NL

1 OF 2
AD
A045036





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A045036

(12)

TECHNICAL REPORT TD-CR-77-4

TIVAR: A COMPUTER PROGRAM FOR PREDICT ENSEMBLE STATISTICS IN A TIME-VARYING MAN/MACHINE CONTROL TASK

U.S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND

Aeroballistics Directorate
Technology Laboratory

15 July 1977

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Redstone Arsenal, Alabama 35809

DMI FORM 1000, 1 APR 77

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DRDMI-TD-CR-77-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TIVAR: A COMPUTER PROGRAM FOR PREDICTING ENSEMBLE STATISTICS IN A TIME-VARYING MAN/MACHINE CONTROL TASK.		5. TYPE OF REPORT & PERIOD COVERED Contractor Technical Report,
		6. PERFORMING ORG. REPORT NUMBER BBN-3464
7. AUTHOR(s) David L. Kleinman, Sheldon Baron and Jeffrey E. Berliner		8. CONTRACT OR GRANT NUMBER(s) DAAH01-76-C-0194
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Proj No. IL362303A214 AMCMS Code No. 632303.2140511
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, US Army Missile R&D Command ATTN: DRDMI-TI Redstone Arsenal, Alabama 35809		12. REPORT DATE 15 July 1977
		13. NUMBER OF PAGES 140
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Commander, US Army Missile R&D Command ATTN: DRDMI-TDW Redstone Arsenal, Alabama 35809		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(12) 141p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

TIVAR
Computer program
Ensemble mean
Standard deviations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report gives the problem formulation, the methods by which system parameters can be changed, the input deck setup, a description of the TIVAR subroutines, and a sample problem with its solution. The report also includes a description of and instructions for using the auxiliary gain computation program GNPROP.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

TABLE OF CONTENTS

1. COMPUTER PROGRAM ABSTRACT

PROGRAM NAME: TIVAR

ORIGINATOR: Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138
(617) 491-1850
David L. Kleinman, Sheldon Baron and Jeffrey E. Berliner

CONTRACT MONITOR: U. S. Army Missile Command
Aeroballistics Directorate
(205) 876-1951
Richard E. Dickson

CONTRACT NUMBER: DAAH01-76-C-0194

PROGRAM ABSTRACT

TIVAR is a highly modular computer program for generating predictions of ensemble mean and standard deviations for a closed-loop manual control system. The system dynamics are assumed linear, but can have arbitrary time variations. The feedback control is generated by the optimal control model of human response, extended to treat time-varying systems with both random and deterministic input disturbances.

The TIVAR program is written in the FORTRAN-IV-EXTENDED computer programming language and is designed for efficient batch operation on a Control Data CDC-6600 computer. Data input to the program is provided on standard punched cards and output is generated via the lineprinter.

In this manual, we give the problem formulation, the methods by which system parameters can be changed, the input deck setup, a description of the TIVAR subroutines, and a sample problem with its solution. The manual also includes a description of and instructions for using the auxiliary gain computation program GNPROP.

2. PROBLEM DESCRIPTION

2.1 System/Display Dynamics

The system being controlled is described by the state-space equation:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ew(t) + Fz(t) \tag{1}$$

where:

$x(t)$ = system state, NX vector, of which NX1 states are associated with input shaping

$u(t)$ = control input, NU vector

$z(t)$ = non-random, deterministic mean or bias inputs, NZ vector

$w(t)$ = independent zero mean white Gaussian noise inputs, NW vector, with covariance:

$$E[w_i(t)\, w_i(\sigma)] = W_i^o(t)\, \delta(t-\sigma) \quad i=1,\ldots,NW \tag{1a}$$

The system parameters, which can be time varying, are:

A = NX by NX state matrix

B = NX by NU control matrix

E = NX by NW noise matrix

F = NX by NZ mean input matrix

The displayed system variables, which includes displayed elements plus their rates of change for man-machine studies, are given by:

$$y(t) = Cx(t) + Du(t) \tag{2}$$

where:

$y(t)$ = displayed information, NY vector

C = NY by NX display matrix for $x(t)$

D = NY by NU display matrix for $u(t)$

2.2 Feedback Control Solution

The system (1) is assumed to be controlled to minimize a quadratic cost functional:

$$J = E[x'(T_f)\, \bar{Q}\, x(T_f) + \frac{1}{Tf}\int_{T_o}^{T_f} (x'q_x x + y'Q_y y + u'Q_u u + \dot{u}'Q_r \dot{u})\, dt] \qquad (3)$$

where T_o and T_f are the initial and final times, respectively, and Q_x, Q_y, Q_u and Q_r are diagonal weighting matrices. In the program, these quantities are read and stored as NX, NY, NU, and NU vectors, respectively. The control law that minimizes J is given by:

$$\dot{u}(t) = -L_c\, \hat{\underline{x}}(t) \qquad (4a)$$

for continuous time implementation, or:

$$\dot{u}_n = -L_d\, \hat{\underline{x}}_n \qquad (4b)$$

for discrete time implementation, wherein $\dot{u}$ is treated as peicewise constant over intervals of length = computer time step. The quantity $\hat{\underline{x}}$ is a "best"estimate of the augmented state $\underline{x} = [x,u]'$.

The NU by (NX+NU) gain matrix L_c or L_d may be input to the program. The logic in TIVAR will convert from one to the other as necessary. Note that in general, L_c and/or L_d will vary with time. An interface program GNPROP, which is described in the Appendix of this manual, provides a means for generating time-varying gains L_d by backwards integration of the Riccati equation. Alternatively, TIVAR contains an option for computing the feedback gains L_d in a suboptimal manner. The assumptions are:

1. $T_f \to \infty$, i.e.,system time constants are small relative to the time-to-go.
2. The terminal weightings, $\bar{Q} = 0$.
3. L_d is computed by solving the steady-state discrete Riccati equation, using the program's present stored values for A, B, C, Q_x, Q_y, Q_u, and Q_r.

The continuous time feedback control law (4a) can be re-written as:

$$\dot{u}(t) = -\left[L_{c1} \mid L_{c2}\right] \begin{bmatrix} \hat{x}(t) \\ \hat{u}(t) \end{bmatrix}$$

or, equivalently:

$$T_n \dot{u} + u = -L_{opt} \hat{x}(t) + v_u(t) \tag{5}$$

where $v_u(t)$ is a "motor-noise"; T_n is a "neuro-motor" time constant. The form of Eq.(5) provides a compatability with earlier work in man-machine systems, dealing primarily with steady-state problems where Q_r is adjusted to give $(T_n)_{ii} \simeq 0.1$ sec typically. If TIVAR is used to compute L_d, the equivalent continuous gains L_c are obtained and printed along with the equivalent L_{opt} and T_n.

2.3 Human Limitations

The human generates $\hat{x}(t)$ on the basis of the delayed and noisy perceived information,

$$y_{pi}(t) = N_i[y_i(t-\tau)] + v_{yi}(t-\tau) \quad i=1,\ldots,NY \tag{6}$$

$\tau =$ human's time delay

$v_y(t) =$ observation or sensor noise

and $N_i(.)$ is the non-linear observation threshold.

$$N_i(x) = \begin{cases} x - a_i & x > a_i \\ 0 & |x| \leq a_i \\ x + a_i & x < -a_i \end{cases} \tag{7}$$

In TIVAR, N_i is replaced by its Random Input Describing function which depends on the mean and variance of the input signal and the threshold a_i. The sensor noise v_{yi}, $i=1,\ldots,NY$ is a zero-mean, white Gaussian noise with covariance:

$$E[v_{yi}(t)\, v'_{yi}(\sigma)] = \frac{Vo_{yi}(t)}{fi(t)} \delta(t-\sigma); \quad i=1,\ldots,NY \tag{8}$$

that contains both an additive and a ratioed component, viz,

$$Vo_{yi}(t) = V_{yi}(t) + \pi \rho_{yi} E[y_i 2(t)] \tag{9}$$

The NY additive noise variances V_{yi} and the NY noise/signal ratios ρ_{yi} (in dB) are inputs to TIVAR. The quantity f_i is the attentional allocation to displayed variable y_i. The f_i are constrained by:

$$f_i > 0$$

$$\sum_{i=1}^{NY} f_i = f_{TOT} = \text{"Workload", or total attention}$$

The neuro-motor interface portion of the model is given by Equation 5. The motor noises $v_{ui}(t)$, $i=1,\ldots,NU$ are zero-mean, white Gaussian noises with covariance

$$E[v_{ui}(t)\ v'_{ui}(\sigma)] = V^{o}_{ui}(t)\ \delta(t-\sigma) \tag{10}$$

that also contains an additive and a ratioed component,

$$V^{o}_{ui}(t) = V_{ui}(t) + \pi\ \rho_{ui}\ E[u_i^{\,2}(t)] \tag{11}$$

The NU additive noise variances V_{ui} and the NU noise/signal ratios ρ_{ui} (in dB) are inputs to TIVAR.

2.4 Nominal or Equilibrium Path

In the previous equations, the state x(t) represented the deviations of the system motion from some nominal path, $X_{nom}(t)$. Similarly, y(t) is the deviations from some nominal $Y_{nom}(t)$. In most applications X_{nom} and Y_{nom} are zero, i.e., X=0 is the equilibrium path. TIVAR can be used to generate a specific nominal path $U_{nom}(t)$, $X_{nom}(t)$, $Y_{nom}(t)$ that meets an NTF terminal condition:

$$H_x\ X(T_f) + H_u\ U(T_f) + c = 0 \tag{12}$$

for the deterministic system:

$$\dot{X}(t) = A\ X(t) + B\ U(t)\ ; \quad [X(T_o),\ U(T_o)] = \text{given} \tag{13}$$

while minimizing the cost functional:

$$J_{nom} = \int_{T_o}^{T_f} \dot{U}'(t)\ Q_r\ \dot{U}(t)\ dt \tag{14}$$

The system equation (13) is the same as Equation (1) less the external disturbances w(t) and z(t). The cost functional weighting Q_r is the same as in Equation (3).

The terminal conditions (12) may be written more compac. ֺ as

$$H\,\underline{X}(T_f) = 0 = [H_x \mid H_u \mid c] \begin{bmatrix} X(T_f) \\ U(T_f) \\ 1 \end{bmatrix} \tag{15}$$

The NTF by (NX+NU+1) matrix H is an input to TIVAR. In computing X_{nom} and the output

$$Y_{nom} = C\,X_{nom} + D\,U_{nom}$$

it is assumed that the matrices A, B, and QR are constant over the interval $[T_O, T_f]$. If at some time t, these quantities change, it is necessary to resolve the optimization with T_o=t and $X_{nom}(t)$ = present value of X_{nom}. If this is not resolved, the terminal conditions (12) will not be satisfied. It is assumed that A, B, QR will remain constant over $[t, T_f]$ when computing X_{nom}.

2.5 Mean and Covariance Propagation Results

TIVAR discretizes the continuous time system equations for the time propagation of the ensemble statistics. The discretization interval Δ is a program input. The ensemble statistics include the mean and standard deviations of any state, output, or control. The mean component $\overline{x}_t = E[x(t)]$ results from:

1. the deterministic input z(t)
2. the nominal path $X_{nom}(t)$

The standard deviation about the mean, σ_x, is a result of:

1. the random input w(t)
2. the human sensor noise $v_y(t)$
3. the human motor noise $v_u(t)$

3. TIME VARIATIONS

In executing TIVAR, any of the input parameters can be changed by the user at any time t. There are two methods for changing parameters. The first is via card inputs; the second is via a user written routine called INTNEW.

3.1 System Codes

In TIVAR, a maximum of 32 system elements may be changed at any given time step. Each element, or parameter, is identified by a unique alphanumeric code and/or an index number. Table 1 defines the system codes used. When a given parameter I is changed at time t, TIVAR sets a flag, IFLAG(I) equal to 1 for one time step. The implication of the parameter change is then addressed via the internal logic in TIVAR. If there has been no parameter change, the internal flags remain at their nominal zero value.

3.2 Changing System Parameters

Parameters may be changed at time t via external or card inputs. The alphanumeric code(s) is used to identify the parameter(s) being changed at a selected time. The input required for each code is given in Section 4.

Parameters can be changed periodically via an internal subroutine INTNEW. The user must supply his own code -- in FORTRAN IV -- to use this option. For any specified code index number, I, the subroutine INTNEW(KEY) is called once every NDT(I) time steps with KEY=I. The 32 values of NDT are inputs to TIVAR. When INTNEW is called, IFLAG(KEY) is set to 1. The user must supply the manner in which the parameters are to be changed. If no FORTRAN code is supplied to update the parameters, no changes are made. Thus, TIVAR assumes that the "new" parameters are identical to the previously existing ones.

The ability to change parameters easily, via user-written code, enables the user to study problems in which:

1. system variables are changing continuously, i.e. eve.y NDT=k time steps.
2. only certain elements in the system matrices are changing with time.
3. system variables are functions of the system ensemble statistics, e.g., nonlinear systems that are being linearized.
4. logic decisions determine how parameters change.

Table 1: PARAMETER CODES IN TIVAR

CODE	INDEX	DESCRIPTION
A	1	System A matrix, Eq. (1)
B	2	System B matrix, Eq. (1)
C	3	Output C matrix, Eq. (2)
D	4	Output D matrix, Eq. (2)
E	5	Noise matrix, E, Eq. (1)
QX	6	State weightings vector, Eq. (3)
QY	7	Output weightings vector, Eq. (3)
QU	8	Control weightings vector, Eq. (3)
QR	9	Control rate weightings vector, Eq. (3)
TD	10	Human's time delay τ, Eq. (6)
PN	11	Variance of random inputs w, Eq. (1a)
MNA	12	Additive motor noise variances, V_u, Eq. (11)
MNR	13	Motor noise ratios, ρ_u, Eq. (11)
SNA	14	Additive sensor noise variances, V_y, Eq. (9)
SNR	15	Sensor noise ratios, ρ_y, Eq. (9)
TH	16	Observational thresholds, a_i, Eq. (7)
ATTN	17	Attentional allocations, f_i, Eq. (8)
CGAIN	18	Continuous time control gains, Eq. (4a)
DGAIN	19	Discretized control gains, Eq. (4b)
PNTVL	20	Time interval for program printouts
NEWG	21	Trigger to compute L_d in program
XMINC	22	Increment mean value of states
XSINC	23	Increment standard deviation of states
F	24	Deterministic input matrix, F, Eq. (1)
XNOM	25	Nominal path initial condition, Eq. (13)
TCR	26	Terminal condition matrix, H, Eq. (15)
INT	27	Transfer control to subroutine INTNEW
DUMMY	28-32	Dummy codes, inactive at present

4. INPUT DECK SETUP

The input deck structure for TIVAR is discussed along with the user-written routines INTNEW and FDOT.

4.1 Control Cards

There are six major control cards required by TIVAR.

Card 1 - Title Information

Column 1: blank

Columns 2-80: alphanumeric title information

Card 2 - Dimension Information, 7I5 Format

Field 1: NX = number of system states

Field 2: NX1= number of noise shaping states

Field 3: NU = number of control inputs ≤ 4

Field 4: NW = number of random noise sources

Field 5: NY = number of displayed outputs

Field 6: NZ = number of deterministic inputs

Field 7: NTF= number of terminal conditions for nominal path trajectory

The size restrictions are:

$$NX + NU \leq NDIM$$

$$NY \leq NDIM$$

where NDIM is the array size in the DIMENSION statements (15).

Card 3 - Time Information, 4E10.0 Format

Field 1: DEL = discrete time step interval (sec)

Field 2: T0 = initial time (sec)

Field 3: TEND = terminal time (sec)

Field 4: TEXT = time extension to propagate covariance TEXT sec past TEND.

Card 4 - Printing/Plotting Control, 3(20I1) Format

Field 1: Cols. 1-20 Print/Plot codes for states 1-NX

Field 2: Cols. 21-40 Print/Plot codes for outputs 1-NY

Field 3: Cols. 41-60 Print/Plot codes for controls 1-NU.

Each column of an associated field corresponds to one state, output or control. A single integer governs the printing or plotting of the results for that variable:

0, or blank = no printing or plotting of the variable

1 = print mean and standard deviation vs. time

3 = plot $\overline{x}$ and $\overline{x} \pm \sigma_x$ vs. time

2 = print <u>and</u> plot ensemble statistics

A maximum of 5 states, 5 outputs and 5 controls may be printed on wide paper. Narrow paper will accomodate only 3 variables per states, outputs or controls.

<u>Cards 5-6</u> - NSTEP(I) for internal time breaks 32I5 Format

The 32 fields are associated with the 32 parameter codes in TIVAR on a one-to-one basis. The I-th field is associated with code I. NSTEP(I) is the frequency (number of time steps) at which subroutine INTNEW is called internally with KEY=I, starting at time T0. Calling INTNEW with KEY=I sets IFLAG(I)=1 for one time step. The actual parameter values must be changed by user-written code. If no code is supplied, the associated parameters retain their previous values.

4.2 External Parameter Cards

The remaining input cards are used to change system parameters via external read-in at specified times. The deck set-up follows a standard form.

<u>Time Card</u> - Cols. 1-4 Alphanumeric TIME

Cols. 11-20 Time of external break, E10.0 Format

<u>Code Card</u> - Cols. 1-5 One of the alphanumeric codes given Table 1.

<u>Parameter Cards</u> - The new parameter values required by the associated code.

The sequence of Code Card followed by new parameter values is repeated for all items that the user wishes to change at the given time. To change parameters at another time, input a new time card with the time of the desired change, followed by a code card, parameter cards, code card, etc. When using external (card) updates, the following rules must be observed:

1. Time breaks must occur in increasing order.
2. The code cards can occur in any order.
3. The parameter cards must immediately follow the associated code card.
4. Parameter cards _must_ be input, as the program expects to read in new values.
5. The last card in the deck is an end card, containing the alphanumeric END in cols. 1-3.

Table 2 gives the required input for each of the active codes, as well as the initial parameter values that are set internally by the program, prior to t=T0.

Data is entered on the cards in 8E10.0 Format, i.e., in floating point fields of 10 columns with a maximum of 8 fields per card. The numbers may be either in fixed-point (decimal) format or in scientific (exponential) format with the exponent right-justified in the field. Matrices are entered one row at a time. If a row contains less than 8 entries, the remaining fields on the card are left blank. If a row contains more than 8 entries, continue on a second card for that row. A new row always begins on a new card. Vectors are enterd in similar 8E10.0 format: the first entry in the first field, the second entry in the second field, etc.

4.3 User Written Routines

The two user written routines are INTNEW(KEY) and FDOT(K,T). The purpose of INTNEW has been discussed earlier as regards parameter variations. The function FDOT(K,T) is used to provide the time history of the deterministic inputs $z_i(t)$. Thus at time t, and for input K, K=1,...,NZ:

TABLE 2: CARD DATA INPUTS AND INITIALIZATION

CODE	INDEX	INPUT DATA	INITIAL VALUE
A	1	A_{ij} ; i=1,...,NX, j=1,...,NX	A=0
B	2	B_{ij} ; i=1,...,NX, j=1,...,NU	B=0
C	3	C_{ij} ; i=1,...,NY, j=1,...,NX	C=0
D	4	D_{ij} ; i=1,...,NY, j=1,...,NU	D=0
E	5	E_{ij} ; i=1,...,NX, j=1,...,NW	E=0
QX	6	Q_{xi} ; i=1,...,NX	$Q_x=0$
QY	7	Q_{yi} ; i=1,...,NY	$Q_y=0$
QU	8	Q_{ui} ; i=1,...,NU	$Q_u=0$
QR	9	Q_{ri} ; i=1,...,NU	$Q_r=0$
TD	10	τ	$\tau=0$
PN	11	W^o_i ; i=1,...,N W	W=0
MNA	12	V_{ui} ; i=1,...,NU	$V_u=0$
MNR	13	ρ_{ui} ; i=1,...,NU	$\rho_u=-\infty$dB
SNA	14	V_{yi} ; i=1,...,NY	$V_y=0$
SNR	15	ρ_{yi} ; i=1,...,NY	$\rho_y=-\infty$dB
TH	16	a_i ; i=1,...,NY	a=0
ATTN	17	f_i ; i=1,...,NY	f=1
CGAIN	18	$(L_c)_{ij}$; i=1,...,NU, j=1,...,NX+NU	$L_c=0$
DGAIN	19	$(L_d)_{ij}$; i=1,...,NU, j=1,...,NX+NU	$L_d=0$
PNTVL	20	h = printout interval	h=0
NEWG	21	no input required	---
XMINC	22	$\delta\bar{X}_i$; i=1,...,NX+NU	$\delta\bar{x}=0$
XSINC	23	$(\delta\sigma^2_x)_{ii}$; i=1,...,NX+NU	$\delta\sigma^2_x=0$
F	24	F_{ij} ; i=1,...,NX, j=1,...,NZ	F=0
XNOM	25	$(X_{nom})_i$; i=1,...,NX+NU	$X_{nom}=0$
TCR	26	H_{ij} ; i=1,...,NTF, j=1,...,NX+NU+1	H=0
INT	27	KEY in I2 Format	---

$$FDOT(K,T) = z_K(T)$$

If NZ=0, i.e., no deterministic inputs, FDOT should return 0.0. The user must supply his own code for FDOT.

4.4 Miscellaneous Comments

The following comments are pertinent to the use of TIVAR.

1. If they occur at the same time instant, external (i.e., card input) updates take precedence over internal updates.

2. Reading control gains L_c or L_d takes precedence over an internal computation request (NEWG).

5. TIVAR SUBROUTINES

The various subroutines that constitute the TIVAR package are discussed briefly, along with the named COMMON blocks.

5.1 Subroutine Descriptions

The following subroutines are used for the covariance propagation program. Their functions are described below briefly.

1. Subroutine TIVAR. Provides the major logic control, dimensioning declarations and initialization for the entire package. Virtually no computation is done in TIVAR. TIVAR reads the 6 control cards.

2. Subroutine UPDATE. Reads input code cards and the subsequent data input cards. Calls INTNEW periodically as per NSTEP.

3. Subroutine INTNEW. User written routine for internal updates.

4. Subroutine GPFBGN. Obtains discrete feedback gains L_d from l_c input, or via internal computations. Converts L_c to l_d and vice-versa as necessary. Outputs equivalent T_N and L_{opt}.

5. Subroutine COVAR. Performs a onetime-step propagation of the vector mean and matrix covariance equations, using the current values of system parameters. Updates matrix one time step. This routine is the heart of the entire package.

6. Subroutine FINAL. Generates the nominal trajectory X_{nom} and Y_{nom}, when $H \neq 0$.

7. Subroutine INFORM. Stores the information for printing/plotting on a disk file for subsequent output. Computes max and min scaling for the plot routine.

8. Subroutine PRINTR. Prints and plots output information as requested on control card 4.

9. Function XGAIN. Computes Random Input describing function for threshold nonlinearity.

10. Function FDOT. User written routine to generate deterministic inputs $z_i(t)$.

11. Library Routines. In addition to the above routines, the program requires many of the subroutines from the linear system library.

5.2 <u>Common Block Usage</u>

Named COMMON blocks are used to store most data arrays and to pass information among the various subroutines. The dimension declarations are given in the primary subprogram TIVAR.

1. /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32), NSTEP(32),
   LPRNT(60), XMAX(60), XMIN(60)

   NCODES = Number of possible system codes in Table 1 (32)
   PINTVL = Printout interval, read in code 20 (PSP)
   ICODES = Alphanumeric code identifiers
   IFLAG = 0 or 1 flags to indicate parameter changes
   NSTEP = frequencies for internal updates, control cards 5-6
   LPRNT = print/plot control read on card 4
   XMAX, XMIN = max and min values for dynamic variables (computed internally)

2. /INOU/ KIN, KOUT, KPUNCH, KDISK

   Logical unit numbers for input/output devices

3. /PLOT1/

   Required input for lineprint plot subroutine

4. /MAIN1/ NDIM, NDIM1, COM1 /MAIN2/ COM2

   Common blocks required for library subroutines. NDIM = dimension of program square arrays.

5. /COMP1/, /COMP2/, /COMP5/

   Common blocks used for internal computations and internal storage of temporary matrices.

6. /INPTS/ BD, SAV, AD

   Discretized system variables

   AD = discrete system augmented matrix = $e^{A_c \delta}$

   BD = discrete system B matrix = $\int_0^{\delta} e^{A_c \sigma} d\sigma \, B_c$

   SAV = storage for the last NU columns of A

7. /INPTX/ N, NX, NX1, NU, AC(1)

   Continuous system state parameters.

   AC = augmented system matrix (codes 1 and 2)

   N = NX + NU.

8. /INPTY/ NY, C(1) /INPTW/ NW, E(1) /INPTWD/ NZ, F(1)

   Input information for display outputs, random inputs and deterministic inputs, respectively.

9. /TIMES/ TIME, DEL, TO, TEND, TEXT, TD, NPRED, EA(1)

   TIME = current value of time, t

   DEL = discrete time step, δ, (card 3)

   TO, TEND, TEXT = initial, final and time extension (card 3)

   TD = time delay τ, NPRED = $[\tau/\delta]$

   EA = $(A_d)^{NPRED} = \exp(A_c \tau)$

10. /WEIGHT/ QX(30), QY(30), QR(30), PSS(1)

    QX, QY, QR = (augmented) state, output and control rate weightings respectively.

    PSS = steady-state Riccati equation solution.

11. /RATIOS/ PU(30), VU(30), PY(30), VY(30), TH(30), GTH(30)

    PU, VU = motor noise ratios, additive components

    PY, VY = observations noise ratios, additive components

    TH, GTH = observation thresholds, RIDF gain

12. /GAINBK/ GAINS(1)

    GAINS = state feedback gains L_d

13. /NOISE/ VUO(30), VYO(30), WO(30), SIGMA(1)

    VUO = total motor noise covariance at time t

    VYO = total observation noise covariance at time t

    WO = random input noise variance

    SIGMA = filter Riccati equation solution at time t

14. /INCRE/ XINC(30), XCINC(30), ATTN(30)

    XINC = mean increment $\delta\overline{x}$

    XCINC = SD increment $\delta\sigma_x$

    ATTN = attentional allocation, f_i

15. /NOMNL/ NTF, XNOM(30), YNOM(30), TCR(1)

    NTF = number of terminal conditions

    XNOM, YNOM = present value of X_{nom}, Y_{nom}

    TCR = terminal condition matrix, H, code 26

16. /COVX/ YBAR(30), YCOV(1)

    XBAR = present value of augmented $\overline{x}(t)$

    XCOV = present state covariance matrix

17. /COVY/ YBAR(30), YCOV(1)

    YBAR = present value of mean output, $\overline{y}(t)$

    YCOV = present output covariance matrix

18. /COVXH/ XHN(30), XHCOV(1)

    XHN = present value of mean state estimate $\hat{\overline{x}}(t)$

    XHCOV = present covariance of state estimate

19. /COVEF/ ENF(30), ENP(30), ECOV(1)

    ENF = present value of mean filtering error, $\overline{e}_f(t)$

    ENP = present value of mean prediction error, $\overline{e}_p(t)$

    ECOV = filter error covariance at time t

20. /COVXHE/ DUM(30), XHECOV(1)

DUM = dummy vector

XHECOV = cross-covariance between e_f and $\hat{x}$ at time t

6. SAMPLE PROBLEM

A sample problem illustrating many of the features of the TIVAR program is given in this section. A description of the problem is presented first, followed by a listing of the user written subroutines, the input data deck, and a listing of the output.

6.1 Sample Problem Description

This problem analyzes an AAA tracking task. The controller tracks the azimuth angle of a target which is executing a level fly-by. The key element illustrated by this problem is the use of the FDOT function to generate the time history of the deterministic input, i.e. the azimuth trajectory of the target.

The controller is explicitly presented with a display of the azimuth sighting error, and is assumed to derive the corresponding error rate. His task is to minimize this error by controlling a set of rate-aided second-order sight dynamics, his control being a hand-crank.

The target being tracked is executing a constant speed straight and level fly-by of 44 sec duration. The range of the target at crossover, R_c, is 3000 ft, and the speed, V, is 733 ft/sec producing a maximum azimuth angular velocity of about 14 deg/sec at crossover. Initially, 22 sec before crossover, the target is 16,126 ft from the crossover point, its azimuth angular position is -79.46 deg, and its azimuth angular velocity is 0.4683 deg/sec.

The system states are defined as follows:

x_1 = target azimuth angular position (degrees)
x_2 = target azimuth angular velocity (deg/sec)
x_3 = sight azimuth angular position (degrees)
x_4 = sight azimuth angular velocity (deg/sec)
x_5 = integral of the control input

It is assumed that the controller employs a "constant velocity" model of the target position. Consequently, the state space equations for the first two states (the deterministic states) are:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = z(t),$$

where z(t), the deterministic input is the azimuth angular <u>acceleration</u> of the target. Thus, z(t) is given by:

$$z(t) = -2(V/R_c)2\frac{D(t)/R_c}{(1 + (D(t)/R_c)^2)^2} \tag{16}$$

where

V = the speed of the target = 733 ft/sec

R_c = the range of the target at crossover = 3000 ft

D(t) = the distacne of the target from the crossover point

$= D_o + Vt = -16,126 + 733t$

The FDOT function computes z(t) according to Eq. 16.

The transfer function relating the sight position to the control input is:

$$\frac{x_3(s)}{u(s)} = \frac{64(s+1)}{s(s^2+12s+64)} = (1+1/s)\frac{64}{s^2+12s+64} \tag{17}$$

Consequently, the state space equations for the last three states (the controllable states) are:

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -64x_3(t) - 12x_4(t) + 64x_5(t) + 64u(t) \tag{18}$$

$$\dot{x}_5 = u(t)$$

The displayed outputs are the azimuth sighting error and error rate and are given by:

$$
\begin{aligned}
y_1(t) &= x_1(t) - x_3(t) \\
y_2(t) &= x_2(t) - x_4(t)
\end{aligned}
\tag{19}
$$

Regarding the human's inherent limitatons, the observation noise to signal ratio (SNR) and motor noise ratio (MNR) are set to the nominal values of -20dB and -25dB, respectively. The perceptual time delay (TD) is set to 0.20 sec. Observational thresholds are set at 0.05 deg for y_1 (corresponding to 1% of the field of view of the gunsight), and 0.025 deg for y_2 (corresponding to a nominal differential threshold for motion).

The control gains, CGAIN, computed by another program, were chosen by setting the cost on error to unity, and adjusting the cost on control rate to produce a neuro-motor lag $_N$ of 0.1 sec.

Finally, since the mean initial states are far from zero, the human's estimator requires some time to settle down. The presence of the human's time delay further compounds this problem. To provide an initializing transition period while the human's estimator settles down, the sample problem is started at t=-6.0 sec, at which time the target is 20,524 ft from the crossover point, its azimuth angular position is -81.684 deg and its azimuth angular velocity is 0.2928 deg/sec. During this initialization period, the human's time delay is set to zero, and printouts are suppressed. At t=0, however, the time delay is set to 0.20 sec, and the printout interval is set to 1 sec.

The states are incremented, XINC, so that the initial angular position and velocity of the target are correct, and the human's state estimates are incremented, XHINC, so that the initial error and error rate are zero.

6.2 User Written Subroutines for the Sample Problem

```
C       TIVP - PROBLEM DEPENDENT SUBPROGRAMS FOR TIVARI
C       INCLUDES:
C        1 - FUNCTION FDOT
C        2 - SUBROUTINE INTNEW
C        3 - SUBROUTINE ADJNOM


        FUNCTION FDOT(K,T)
C       SPECIFIES THE TIME HISTORY OF THE DETERMINISTIC INPUT
C       PROBLEM DEPENDENT - SUPPLIED BY THE USER
        DATA
     1  X0, Y0, V0 /-16126.0, 3000.0, 733.0/,
     2  R /57.296/
        X=X0+V0*T
        A=X/Y0
        B=1.0+A*A
        C=(V0/Y0)/B
        D=-2.0*A*C*C
        FDOT=R*D
        RETURN
        END


        SUBROUTINE INTNEW(KEY)
C       PERFORMS INTERNAL UPDATES
C       MUST BE SUPPLIED BY USER - PROBLEM DEPENDENT
        COMMON
     1  /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32)
     1          NSTEP(32)
        IFLAG(KEY)=1
        RETURN
        END


        SUBROUTINE ADJNOM
C       ADJUSTS OUTPUTS FOR NOMINAL PATH
C       PROBLEM DEPENDENT
        COMMON
     E  /TIMES/  TIME
     M  /COVY/   YBAR(30)
     Q  /NOMNL/  NTF, XNOM(30), YNOM(30), TCR(15,16)
1       CONTINUE
        RETURN
        END
```

6.3 Input Deck for the Sample Problem

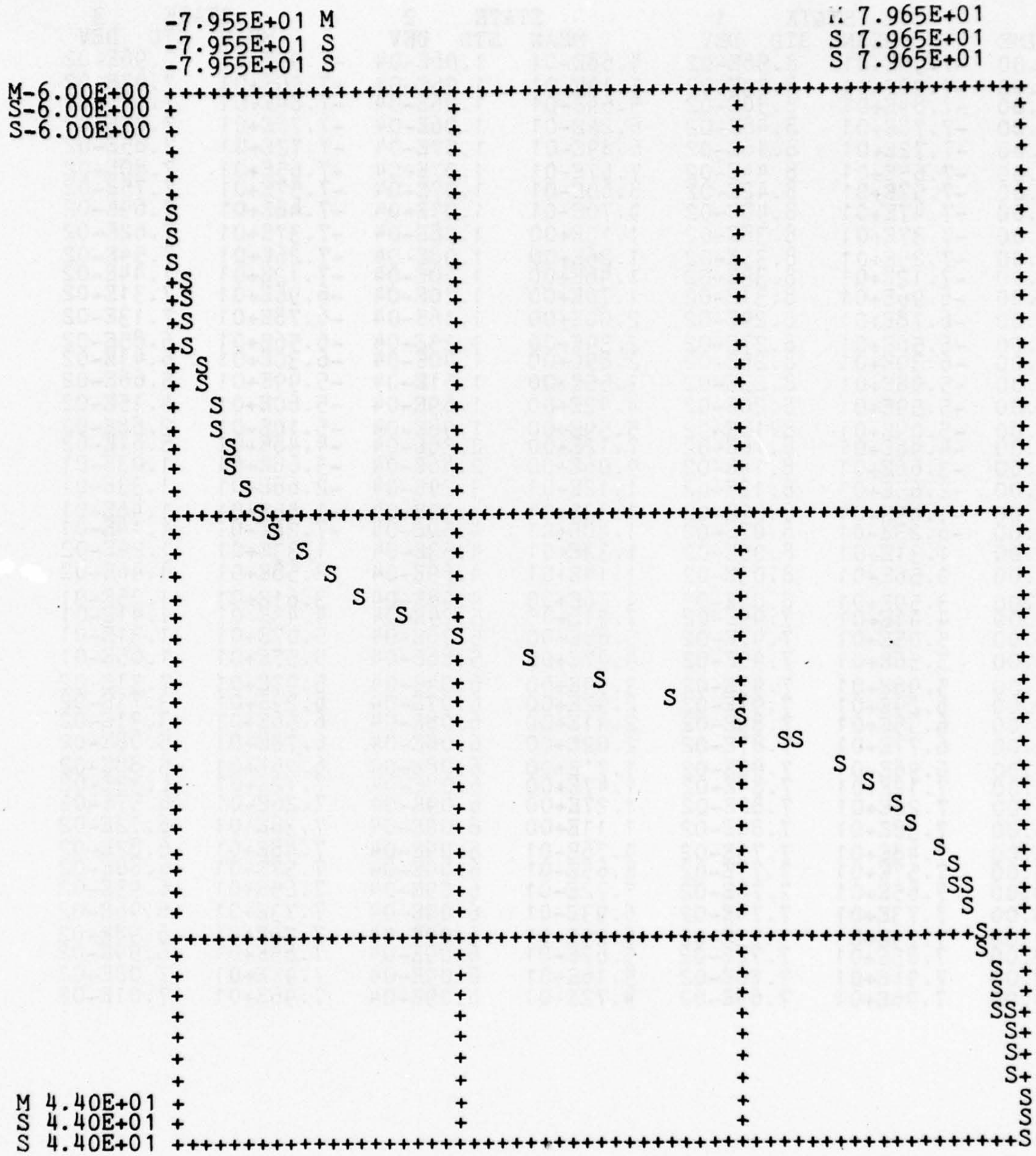
```
P-I-D CONTROLLER.  COVARIANCE PROPAGATION
    5    2    1    0    2    1    0
       0.05         -6.0      44.0         0.0
22200               22                     2
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
A
       0.0          1.0        0.0         0.0         0.0
       0.0          0.0        0.0         0.0         0.0
       0.0          0.0        0.0         1.0         0.0
       0.0          0.0      -64.0       -12.0        64.0
       0.0          0.0        0.0         0.0         0.0
B
       0.0
       0.0
       0.0
      64.0
       1.0
F
       0.0
       1.0
       0.0
       0.0
       0.0
C
       1.0          0.0       -1.0         0.0         0.0
       0.0          1.0        0.0        -1.0         0.0
D
       0.0
       0.0
MNR
     -25.0
SNR
     -20.0        -20.0
TH
      0.05        0.025
TD
      0.00
CGAIN
    -15.39       -3.303      6.359      0.6404       9.034
     10.00
XMINC
   -81.684       0.2928    -81.684      0.2928         0.0
       0.0
PNTVL
       6.0
TIME                0.0
TD
      0.20
PNTVL
       1.0
END
```

6.4 Output Listing for the Sample Problem

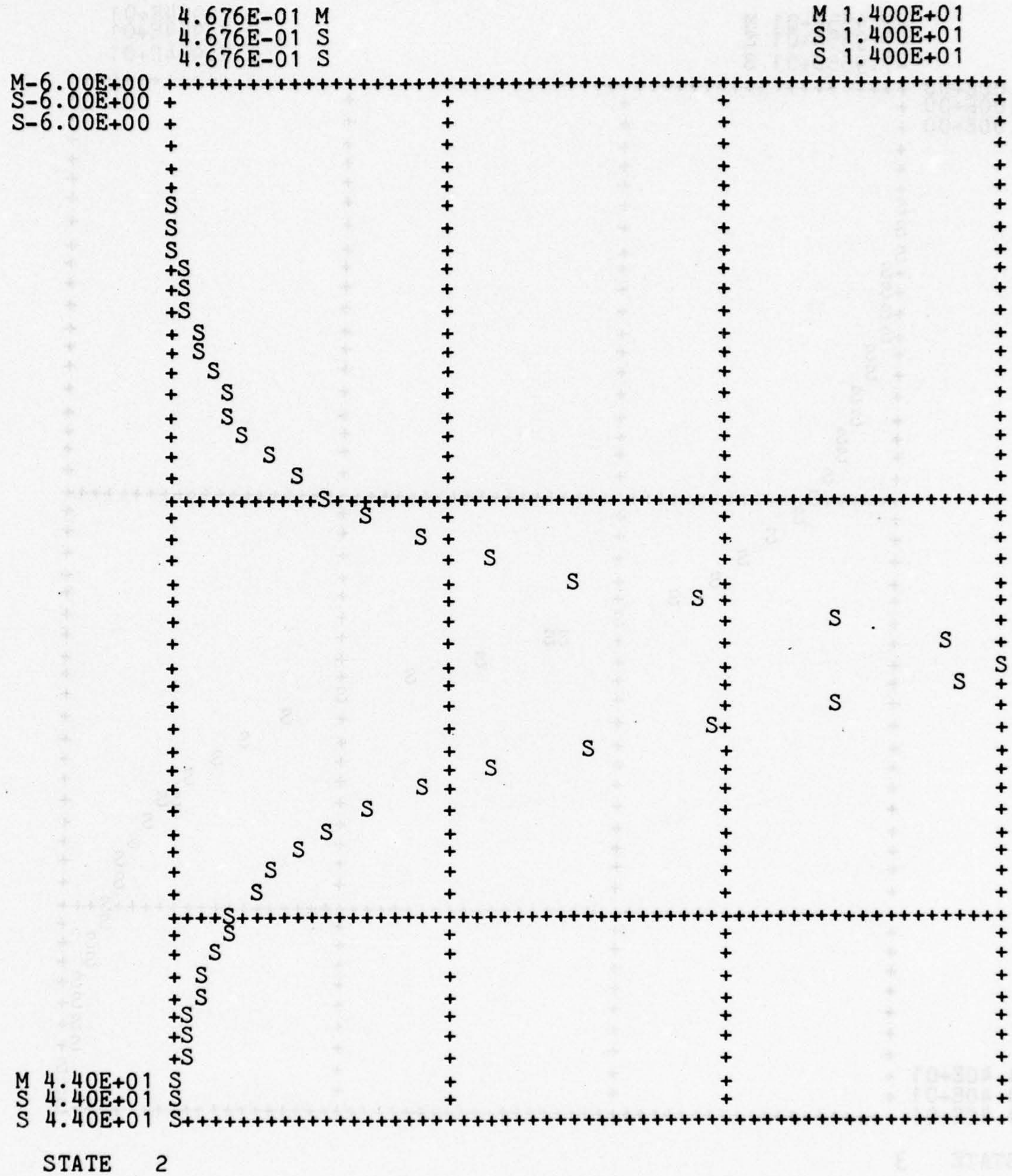
```
P-I-D CONTROLLER
 2-Dec-76   14:28

DYNAMICS READ FROM: PIDTI.DYN
NO. OF TOTAL SYSTEM STATES     5
NO. OF NOISE SHAPING STATES    2
NO. OF CONTROL SYSTEM INPUTS   1
NO. OF RANDOM NOISE SOURCES    1
NO. OF DISPLAYED OUTPUTS       2
NO. OF DETERMINISTIC INPUTS    1
NO. OF TERMINAL CONDITIONS     0

INTEGRATION TIME STEP =        0.050
INITIAL TIME =         -6.000
TERMINAL TIME =        44.000
EXTENSION =             0.000

INTERNAL TIME BREAKS  INDEX  CODE  NDT

EXTERNAL UPDATE AT TIME   -6.000    CODE  F

 F    MATRIX:
    0.
    1.000E+00
    0.
    0.
    0.

EXTERNAL UPDATE AT TIME   -6.000    CODE  MNR

PU    VECTOR:
   -2.500E+01

EXTERNAL UPDATE AT TIME   -6.000    CODE  SNR

PY    VECTOR:
   -2.000E+01   -2.000E+01

EXTERNAL UPDATE AT TIME   -6.000    CODE  TH

TH    VECTOR:
    5.000E-02    2.500E-02

EXTERNAL UPDATE AT TIME   -6.000    CODE  TD
TD   =  0.

EXTERNAL UPDATE AT TIME   -6.000    CODE  CGAIN

CGAIN MATRIX:
   -1.539E+01   -3.303E+00    6.359E+00    6.404E-01    9.034E+00
    1.000E+01

EXTERNAL UPDATE AT TIME   -6.000    CODE  XMINC

XMINC VECTOR:
   -8.168E+01    2.928E-01   -8.168E+01    2.928E-01    0.
    0.

EXTERNAL UPDATE AT TIME   -6.000    CODE  PNTVL
PNTVL=  6.000E+00
EQUIVALENT DISCRETE GAINS GENERATED:
```

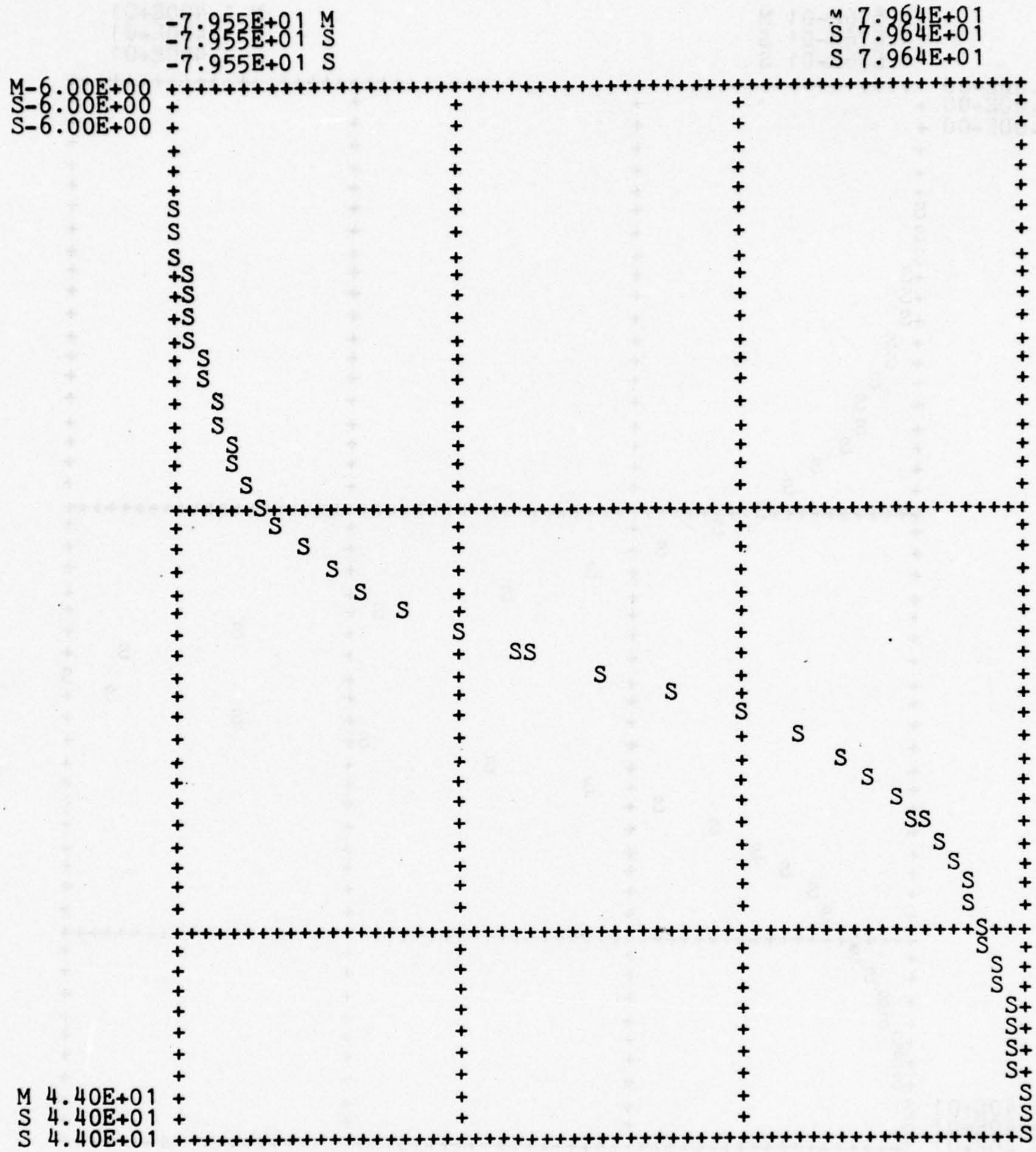
```
DGAIN MATRIX:
   -1.186E+01    -2.867E+00     4.083E+00     4.556E-01     7.782E+00
    8.732E+00

EXTERNAL UPDATE AT TIME     0.000     CODE  TD
TD    =  2.000E-01

EXTERNAL UPDATE AT TIME     0.000     CODE  PNTVL
PNTVL=  1.000E+00
```

	STATE 1		STATE 2		STATE 3	
TIME	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
0.00	-7.95E+01	8.56E-02	4.68E-01	1.06E-04	-7.95E+01	7.96E-02
1.00	-7.90E+01	8.53E-02	5.14E-01	1.06E-04	-7.90E+01	7.93E-02
2.00	-7.84E+01	8.50E-02	5.64E-01	1.06E-04	-7.84E+01	7.92E-02
3.00	-7.78E+01	8.48E-02	6.22E-01	1.06E-04	-7.78E+01	7.89E-02
4.00	-7.72E+01	8.46E-02	6.89E-01	1.07E-04	-7.72E+01	7.85E-02
5.00	-7.65E+01	8.44E-02	7.67E-01	1.07E-04	-7.65E+01	7.80E-02
6.00	-7.57E+01	8.42E-02	8.60E-01	1.07E-04	-7.57E+01	7.75E-02
7.00	-7.47E+01	8.40E-02	9.70E-01	1.07E-04	-7.48E+01	7.69E-02
8.00	-7.37E+01	8.38E-02	1.10E+00	1.08E-04	-7.37E+01	7.62E-02
9.00	-7.25E+01	8.35E-02	1.26E+00	1.08E-04	-7.26E+01	7.54E-02
10.00	-7.12E+01	8.33E-02	1.46E+00	1.10E-04	-7.12E+01	7.44E-02
11.00	-6.96E+01	8.31E-02	1.70E+00	1.10E-04	-6.96E+01	7.31E-02
12.00	-6.78E+01	8.29E-02	2.00E+00	1.16E-04	-6.78E+01	7.13E-02
13.00	-6.56E+01	8.27E-02	2.39E+00	1.23E-04	-6.56E+01	6.85E-02
14.00	-6.30E+01	8.25E-02	2.89E+00	1.40E-04	-6.30E+01	6.41E-02
15.00	-5.98E+01	8.23E-02	3.55E+00	1.51E-04	-5.99E+01	5.66E-02
16.00	-5.59E+01	8.20E-02	4.42E+00	1.69E-04	-5.60E+01	4.15E-02
17.00	-5.09E+01	8.18E-02	5.59E+00	1.98E-04	-5.10E+01	2.66E-02
18.00	-4.46E+01	8.16E-02	7.12E+00	2.36E-04	-4.48E+01	6.81E-02
19.00	-3.66E+01	8.14E-02	9.06E+00	2.46E-04	-3.68E+01	1.03E-01
20.00	-2.65E+01	8.12E-02	1.12E+01	3.29E-04	-2.68E+01	1.33E-01
21.00	-1.43E+01	8.09E-02	1.32E+01	4.32E-04	-1.45E+01	1.46E-01
22.00	-6.23E-01	8.07E-02	1.40E+01	4.50E-04	-7.26E-01	1.14E-01
23.00	1.31E+01	8.05E-02	1.33E+01	4.53E-04	1.33E+01	1.29E-02
24.00	2.56E+01	8.03E-02	1.14E+01	4.69E-04	2.58E+01	8.46E-02
25.00	3.59E+01	8.01E-02	9.16E+00	4.98E-04	3.61E+01	1.25E-01
26.00	4.41E+01	7.99E-02	7.21E+00	5.29E-04	4.43E+01	1.41E-01
27.00	5.05E+01	7.97E-02	5.65E+00	5.70E-04	5.07E+01	1.31E-01
28.00	5.56E+01	7.95E-02	4.47E+00	5.86E-04	5.57E+01	1.05E-01
29.00	5.96E+01	7.93E-02	3.59E+00	6.04E-04	5.97E+01	7.31E-02
30.00	6.29E+01	7.91E-02	2.92E+00	6.07E-04	6.29E+01	3.71E-02
31.00	6.55E+01	7.89E-02	2.41E+00	6.08E-04	6.56E+01	3.31E-02
32.00	6.77E+01	7.87E-02	2.02E+00	6.08E-04	6.78E+01	5.08E-02
33.00	6.96E+01	7.86E-02	1.71E+00	6.08E-04	6.96E+01	5.88E-02
34.00	7.12E+01	7.84E-02	1.47E+00	6.08E-04	7.12E+01	6.32E-02
35.00	7.26E+01	7.82E-02	1.27E+00	6.09E-04	7.26E+01	6.57E-02
36.00	7.38E+01	7.80E-02	1.11E+00	6.09E-04	7.38E+01	6.72E-02
37.00	7.48E+01	7.79E-02	9.76E-01	6.09E-04	7.48E+01	6.82E-02
38.00	7.57E+01	7.77E-02	8.65E-01	6.09E-04	7.57E+01	6.88E-02
39.00	7.65E+01	7.76E-02	7.72E-01	6.09E-04	7.65E+01	6.93E-02
40.00	7.73E+01	7.74E-02	6.93E-01	6.09E-04	7.73E+01	6.96E-02
41.00	7.79E+01	7.73E-02	6.25E-01	6.09E-04	7.79E+01	6.98E-02
42.00	7.85E+01	7.72E-02	5.67E-01	6.09E-04	7.85E+01	6.99E-02
43.00	7.91E+01	7.70E-02	5.16E-01	6.09E-04	7.91E+01	7.00E-02
44.00	7.96E+01	7.69E-02	4.72E-01	6.09E-04	7.96E+01	7.01E-02


-7.955E+01 M
-7.955E+01 S
-7.955E+01 S
M 7.965E+01
S 7.965E+01
S 7.965E+01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01

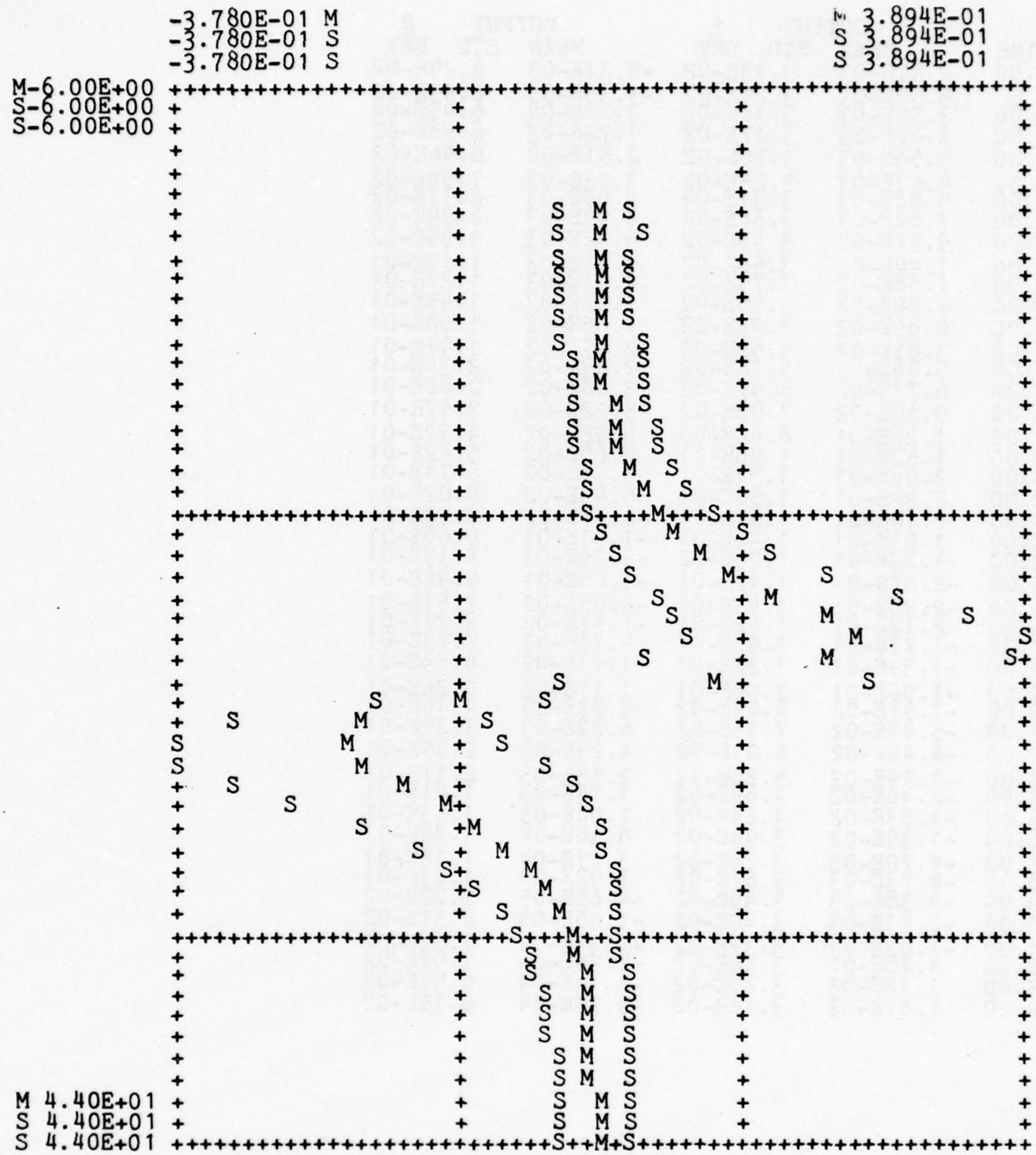
STATE 1


4.676E-01 M
4.676E-01 S
4.676E-01 S
M 1.400E+01
S 1.400E+01
S 1.400E+01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01
STATE 2

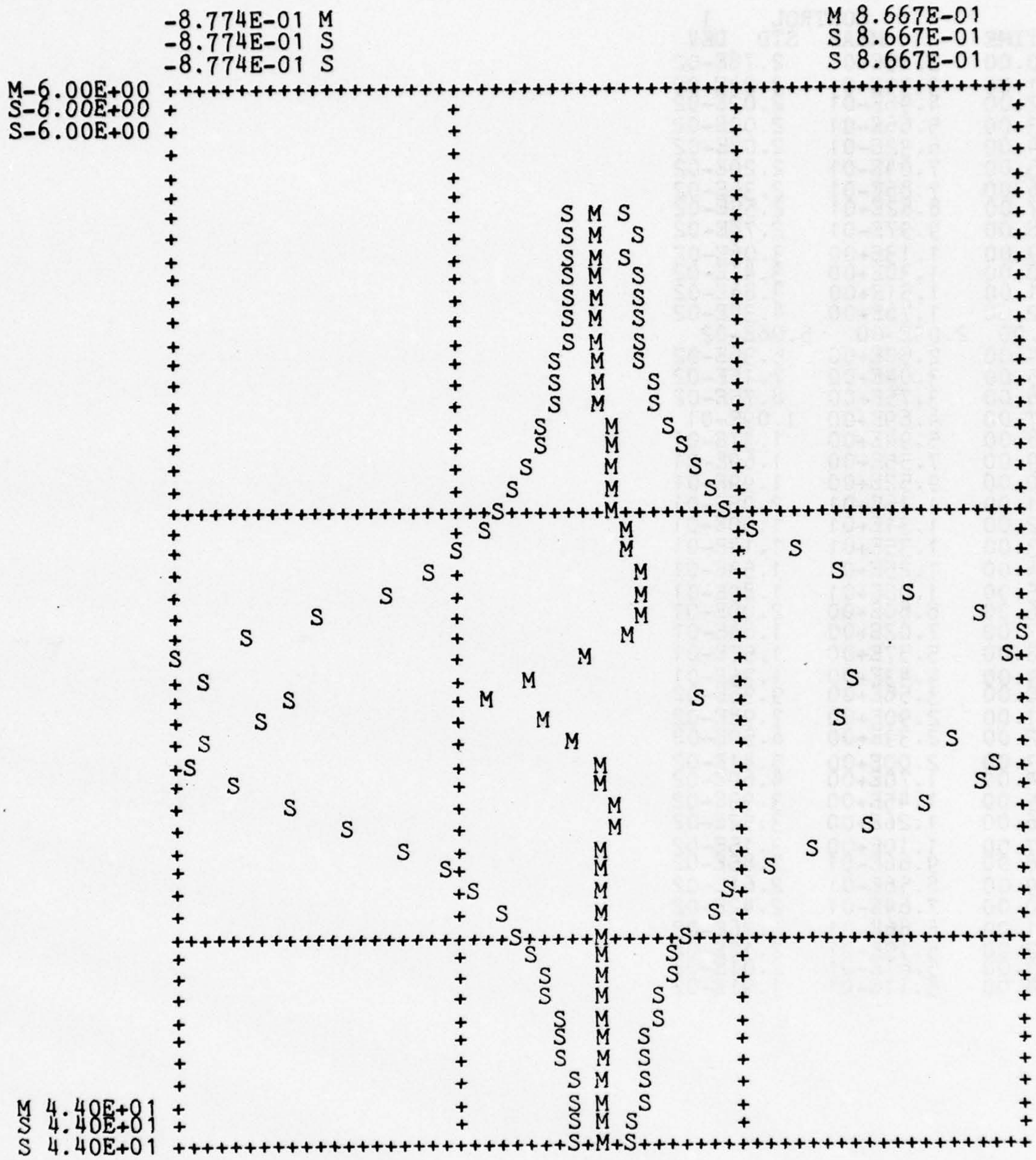

-7.955E+01 M
-7.955E+01 S
-7.955E+01 S
M 7.964E+01
S 7.964E+01
S 7.964E+01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01

STATE 3

TIME	OUTPUT 1 MEAN	OUTPUT 1 STD DEV	OUTPUT 2 MEAN	OUTPUT 2 STD DEV
0.00	2.47E-03	3.13E-02	-8.11E-03	6.29E-02
1.00	5.50E-03	3.29E-02	-1.55E-03	6.93E-02
2.00	3.50E-03	3.18E-02	3.78E-04	6.45E-02
3.00	3.20E-03	3.17E-02	1.72E-03	6.58E-02
4.00	3.66E-03	3.20E-02	2.51E-03	6.96E-02
5.00	4.57E-03	3.25E-02	3.23E-03	7.50E-02
6.00	5.88E-03	3.31E-02	4.00E-03	8.17E-02
7.00	7.62E-03	3.38E-02	4.99E-03	8.99E-02
8.00	9.91E-03	3.48E-02	6.22E-03	9.98E-02
9.00	1.29E-02	3.60E-02	7.81E-03	1.12E-01
10.00	1.68E-02	3.75E-02	9.93E-03	1.27E-01
11.00	2.20E-02	3.95E-02	1.27E-02	1.45E-01
12.00	2.89E-02	4.23E-02	1.65E-02	1.68E-01
13.00	3.81E-02	4.62E-02	2.17E-02	1.97E-01
14.00	5.06E-02	5.18E-02	2.89E-02	2.34E-01
15.00	6.77E-02	5.97E-02	3.88E-02	2.82E-01
16.00	9.10E-02	7.08E-02	5.22E-02	3.47E-01
17.00	1.22E-01	8.60E-02	6.92E-02	4.32E-01
18.00	1.62E-01	1.06E-01	8.68E-02	5.42E-01
19.00	2.06E-01	1.31E-01	9.35E-02	6.74E-01
20.00	2.34E-01	1.56E-01	6.43E-02	8.02E-01
21.00	2.10E-01	1.67E-01	-2.47E-02	8.53E-01
22.00	1.03E-01	1.40E-01	-1.51E-01	6.60E-01
23.00	-1.23E-01	7.95E-02	-2.26E-01	4.19E-01
24.00	-2.07E-01	1.17E-01	-1.08E-01	5.98E-01
25.00	-2.30E-01	1.48E-01	-5.08E-02	7.61E-01
26.00	-2.12E-01	1.62E-01	-1.47E-02	8.27E-01
27.00	-1.77E-01	1.53E-01	4.36E-03	7.73E-01
28.00	-1.39E-01	1.32E-01	1.11E-02	6.54E-01
29.00	-1.06E-01	1.08E-01	1.11E-02	5.26E-01
30.00	-7.91E-02	8.74E-02	8.83E-03	4.16E-01
31.00	-5.89E-02	7.16E-02	6.29E-03	3.29E-01
32.00	-4.40E-02	6.01E-02	4.23E-03	2.65E-01
33.00	-3.29E-02	5.20E-02	2.73E-03	2.17E-01
34.00	-2.46E-02	4.64E-02	1.73E-03	1.81E-01
35.00	-1.83E-02	4.24E-02	1.06E-03	1.53E-01
36.00	-1.35E-02	3.96E-02	6.36E-04	1.32E-01
37.00	-9.70E-03	3.76E-02	3.27E-04	1.16E-01
38.00	-6.73E-03	3.61E-02	1.02E-04	1.03E-01
39.00	-4.38E-03	3.49E-02	-4.65E-05	9.22E-02
40.00	-2.51E-03	3.40E-02	-1.55E-04	8.37E-02
41.00	-1.02E-03	3.32E-02	-2.33E-04	7.66E-02
42.00	1.76E-04	3.26E-02	-2.76E-04	7.07E-02
43.00	1.12E-03	3.20E-02	-3.18E-04	6.57E-02
44.00	1.87E-03	3.16E-02	-3.70E-04	6.15E-02

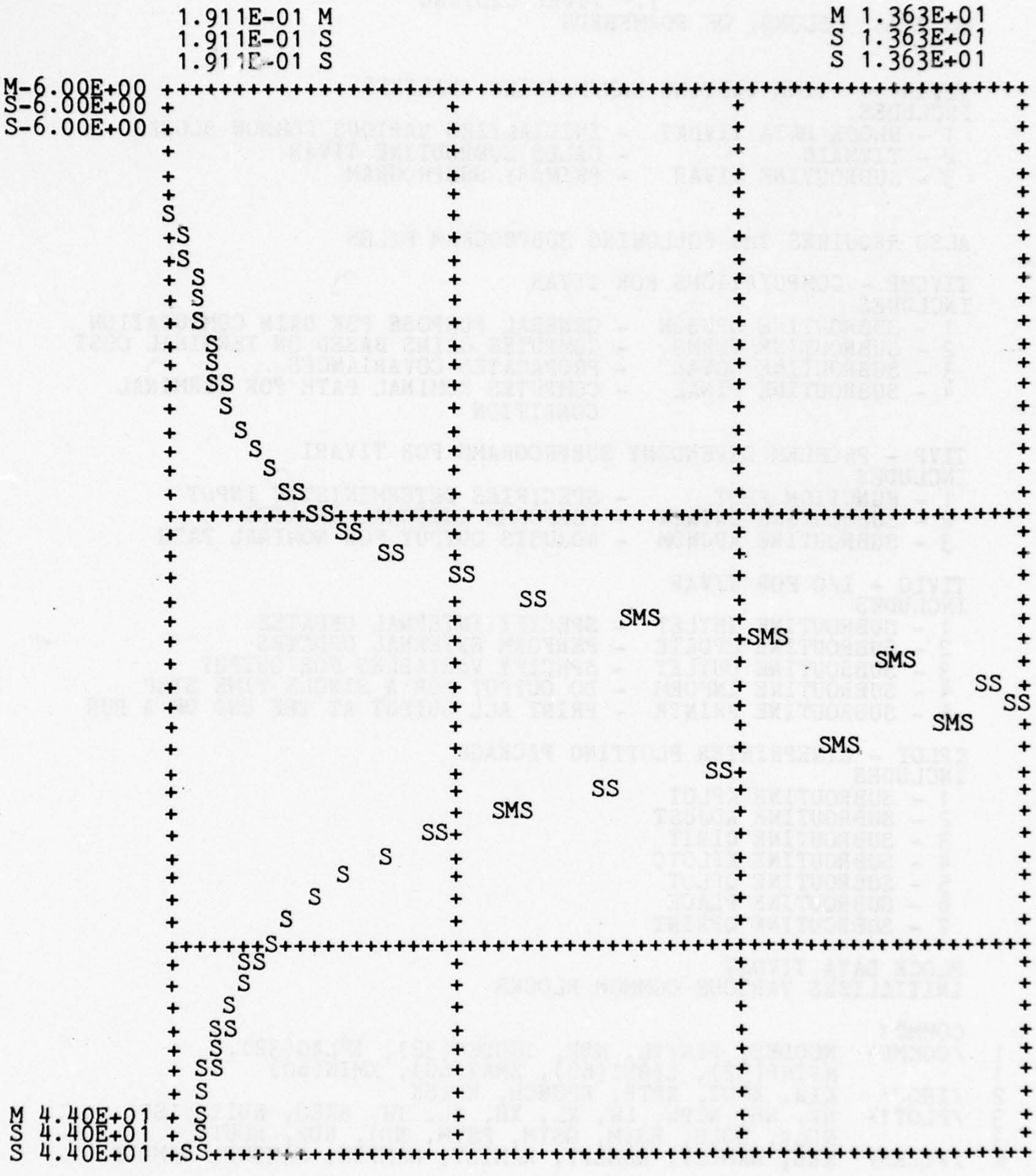

-3.780E-01 M
-3.780E-01 S
-3.780E-01 S
M 3.894E-01
S 3.894E-01
S 3.894E-01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01

OUTPUT 1


-8.774E-01 M
-8.774E-01 S
-8.774E-01 S
M 8.667E-01
S 8.667E-01
S 8.667E-01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01

OUTPUT 2

	CONTROL	1
TIME	MEAN	STD DEV
0.00	2.19E-01	2.78E-02
1.00	3.99E-01	2.24E-02
2.00	4.96E-01	2.02E-02
3.00	5.66E-01	2.00E-02
4.00	6.32E-01	2.08E-02
5.00	7.04E-01	2.20E-02
6.00	7.86E-01	2.36E-02
7.00	8.82E-01	2.55E-02
8.00	9.97E-01	2.78E-02
9.00	1.13E+00	3.06E-02
10.00	1.30E+00	3.41E-02
11.00	1.51E+00	3.84E-02
12.00	1.76E+00	4.38E-02
1.00	2.09E+00	5.06E-02
14.00	2.50E+00	5.96E-02
15.00	3.04E+00	7.15E-02
16.00	3.75E+00	8.75E-02
17.00	4.69E+00	1.09E-01
18.00	5.94E+00	1.37E-01
19.00	7.55E+00	1.69E-01
20.00	9.52E+00	1.99E-01
21.00	1.16E+01	2.05E-01
22.00	1.31E+01	1.50E-01
23.00	1.35E+01	1.14E-01
24.00	1.25E+01	1.53E-01
25.00	1.08E+01	1.89E-01
26.00	8.80E+00	2.00E-01
27.00	7.02E+00	1.84E-01
28.00	5.57E+00	1.55E-01
29.00	4.43E+00	1.25E-01
30.00	3.56E+00	9.95E-02
31.00	2.90E+00	7.98E-02
32.00	2.39E+00	6.50E-02
33.00	2.00E+00	5.41E-02
34.00	1.70E+00	4.60E-02
35.00	1.45E+00	3.99E-02
36.00	1.26E+00	3.52E-02
37.00	1.10E+00	3.15E-02
38.00	9.66E-01	2.85E-02
39.00	8.56E-01	2.61E-02
40.00	7.64E-01	2.42E-02
41.00	6.86E-01	2.26E-02
42.00	6.19E-01	2.12E-02
43.00	5.61E-01	2.01E-02
44.00	5.11E-01	1.91E-02


1.911E-01 M
1.911E-01 S
1.911E-01 S
M 1.363E+01
S 1.363E+01
S 1.363E+01
M-6.00E+00
S-6.00E+00
S-6.00E+00
M 4.40E+01
S 4.40E+01
S 4.40E+01

CONTROL 1

7. TIVAR LISTING

```
C       NO TABS, COLONS, OF FORMFEEDS

C       TIVAR -  TIME VARYING MAN/MACHINE ANALYZER
C       INCLUDES
C        1 - BLOCK DATA TIVDAT   - INITIALIZES VARIOUS COMMON BLOCKS
C        2 - TIVMAIN             - CALLS SUBROUTINE TIVAR
C        3 - SUBROUTINE TIVAR    - PRIMARY SUBPROGRAM

C       ALSO REQUIRES THE FOLLOWING SUBPROGRAM FILES

C       TIVCMP - COMPUTATIONS FOR TIVAR
C       INCLUDES
C        1 - SUBROUTINE GPFBGN   - GENERAL PURPOSE FBK GAIN COMPUTATION
C        2 - SUBROUTINE TERMG    - COMPUTES GAINS BASED ON TERMINAL COST
C        3 - SUBROUTINE COVAR    - PROPAGATES COVARIANCES
C        4 - SUBROUTINE FINAL    - COMPUTES NOMINAL PATH FOR TERMINAL
C                                  CONDITION

C       TIVP - PROBLEM DEPENDENT SUBPROGRAMS FOR TIVARI
C       INCLUDES
C        1 - FUNCTION FDOT       - SPECIFIES DETERMINISTIC INPUT
C        2 - SUBROUTINE INTNEW   - PERFORMS INTERNAL UPDATES
C        3 - SUBROUTINE ADJNOM   - ADJUSTS OUTPUT FOR NOMINAL PATH

C       TIVIO - I/O FOR TIVAR
C       INCLUDES
C        1 - SUBROUTINE INTLET   - SPECIFY INTERNAL UPDATES
C        2 - SUBROUTINE UPDATE   - PERFORM EXTERNAL UPDATES
C        3 - SUBROUTINE OUTLET   - SPECIFY VARIABLES FOR OUTPUT
C        4 - SUBROUTINE INFORM   - DO OUTPUT FOR A SINGLE TIME STEP
C        5 - SUBROUTINE PRINTR   - PRINT ALL OUTPUT AT THE END OF A RUN

C       KPLOT - LINEPRINTER PLOTTING PACKAGE
C       INCLUDES
C        1 - SUBROUTINE KPLOT
C        2 - SUBROUTINE ADJUST
C        3 - SUBROUTINE QINIT
C        4 - SUBROUTINE KPLOTC
C        5 - SUBROUTINE QPLOT
C        6 - SUBROUTINE PLACE
C        7 - SUBROUTINE QPRINT
C
        BLOCK DATA TIVDAT
C       INITIALIZES VARIOUS COMMON BLOCKS

        COMMON
     1  /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1           NSTEP(32), LPRNT(60), XMAX(60), XMIN(60)
     2  /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     3  /PLOT1/  NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     3           NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT
     R  /FILES/  KKB, NAMCOV, NAMLPT, NAMINT, NAMDYN, NAMTMP, NAMBUF

        DATA
     1    NH, NXES, NDIR, NGLV, NGLH, BSYM, GSYM
     1  /101,    1,   10,   20,   20,  1H+,  1H+/,
     2  NOUT, ND1, NCODES, NV
     2  /  6,   1,     32, 51/,
     3  KIN, KOUT, KPTR, KPUNCH, KDISK
     3  / 5,    6,    6,      7,     8/,
     4  (ICODES(I), I=1,32)
     4  /1HA, 1HB, 1HC, 1HD, 1HE, 2HQX, 2HQY, 2HQU, 2HQR, 2HTD,
     4   2HPN, 3HMNA, 3HMNR, 3HSNA, 3HSNR, 2HTH, 4HATTN, 5HCGAIN,
```

```
     4    5HDGAIN, 5HPNTVL, 4HNEWG, 5HXMINC, 5HXSINC, 1HF, 4HXNOM,
     4    3HTCR, 3HINT, 3HQXT, 3HQYT, 3*5HDUMMY/
          END
```

```
      PROGRAM TIVMAIN
    1 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
    2  PUNCH,    DISK, TAPE7=PUNCH,    TAPE8=DISK)
C     MAIN PROGRAM
C     CALLS SUBROUTINE TIVAR

      CALL TIVAR
      END
```

```
      SUBROUTINE TIVAR
C     PRIMARY SUBPROGRAM
      DIMENSION
     1 WORK(15,15), TITLE(8)
      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1          NSTEP(32), LPRNT(60), XMAX(60), XMIN(60)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     3 /PLOT1/  NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     3          NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT
     4 /MAIN1/  NDIM, NDIM1, COM1(15,15)
     5 /MAIN2/  COM2(15,15)
     6 /COMP1/  U(15,15)
     7 /COMP3/  V(15,15)
     8 /COMP5/  EG(15,15)
     9 /INPTS/  BD(15,4), SAV(15,4), A(15,15)
     A /INPTX/  N, NX, NX1, NU, AA(15,15)
     B /INPTY/  NY, C(15,15)
     C /INPTW/  NW, E(15,15)
     D /INPTWD/ NWD, F(15,15)
     E /TIMES/  TIME, DEL, TO, TEND, TEXT, TD, NPRED, EA(15,15)
     F /WEIGHT/ QX(30), QY(30), QR(30), PSS(15,15)
      COMMON
     G /TCOST/  QXT(30), QYT(30), PINV(15,15)
     H /NOISE/  VUO(30), VYO(30), WO(30), SIGMA(15,15)
     I /RATIOS/ PU(30), VU(30), PY(30), VY(30), TH(30), GTH(30)
     J /GAINBK/ GAINS(15,15)
     K /INCRE/  XMINC(30), XSINC(30), ATTN(30)
     L /COVX/   XBAR(30), XCOV(15,15)
     M /COVY/   YBAR(30), YCOV(15,15)
     N /COVXHE/ DUM(30), XHECOV(15,15)
     O /COVXH/  XHN(30), XHCOV(15,15)
     P /COVEF/  ENF(30), ENP(30), ECOV(15,15)
     Q /NOMNL/  NTF, XNOM(30), YNOM(30), TCR(15,16)

C     SET NDIM
      NDIM=15
      NDIM1=NDIM+1

C     ZERO THE VECTORS AND MATRICES
1     NPRED=0
      TD=0.0
      REWIND KDISK
      DO 10 I=1,30
      XMINC(I)=0.0
      XSINC(I)=0.0
      QX(I)=0.0
      QY(I)=0.0
      QR(I)=0.0
      WO(I)=0.0
      TH(I)=0.0
      ATTN(I)=1.0
      XBAR(I)=0.0
      YBAR(I)=0.0
      XHN(I)=0.0
      ENF(I)=0.0
      ENP(I)=0.0
      XNOM(I)=0.0
      YNOM(I)=0.0
      PU(I)=0.0
      VU(I)=0.0
```

```
      PY(I)=0.0
      VY(I)=0.0
10    CONTINUE
      DO 12 I=1,NDIM
      DO 12 J=1,NDIM
      TCR(I,J)=0.0
      SIGMA(I,J)=0.0
      F(I,J)=0.0
      C(I,J)=0.0
      E(I,J)=0.0
      AA(I,J)=0.0
      GAINS(I,J)=0.0
      XCOV(I,J)=0.0
      YCOV(I,J)=10000.0
      XHECOV(I,J)=0.0
      XHCOV(I,J)=0.0
      ECOV(I,J)=1.0E-16
12    CONTINUE
C     SPECIFY THE TITLE
      READ (KIN,1120), (TITLE(I), I=1,8)
      IF (EOF(KIN)) 400,50
1120  FORMAT (8A10)
50    CALL PAGEFD(KPTR,1)
      WRITE (KPTR,1125), (TITLE(I), I=1,8)
1125  FORMAT (/,1H ,8A10)
      CALL DAYTIM(KPTR)
      CALL LINEFD(KPTR,2)
C     GET THE PROBLEM DIMENSIONS
      READ (KIN,1160) NX, NX1, NU, NW, NY, NWD, NTF
1160  FORMAT (16I5)
      WRITE (KPTR,1180) NX, NX1, NU, NW, NY, NWD, NTF
1180  FORMAT (28H NO. OF TOTAL SYSTEM STATES ,I3,/,
     1 29H NO. OF NOISE SHAPING STATES ,I2,/,
     2 29H NO. OF CONTROL SYSTEM INPUTS,I2,/,
     3 29H NO. OF RANDOM NOISE SOURCES ,I2,/,
     4 29H NO. OF DISPLAYED OUTPUTS ,I2,/,
     5 29H NO. OF DETERMINISTIC INPUTS ,I2,/,
     6 29H NO. OF TERMINAL CONDITIONS ,I2,/,
     7 1H )
C     SPECIFY DEL, TO, TEND, AND TEXT
      READ (KIN,1200) DEL, TO, TEND, TEXT
1200  FORMAT (8E10.0)
      TEND=IFIX((TEND-T0+0.0001)/DEL)*DEL+T0
      WRITE (KPTR,1225) DEL, TO, TEND, TEXT
1225  FORMAT (25H INTEGRATION TIME STEP = ,F10.3,/,
     1 18H INITIAL TIME =    ,F10.3,/,
     2 18H TERMINAL TIME =  ,F10.3,/,
     3 18H TIME EXTENSION = ,F10.3,/,
     4 1H )
C     IDENTIFY VARIABLES FOR OUTPUT
      CALL OUTLET
      NSP=1
      DO 59 I=1,60
      IF (LPRNT(I) .GT. 0) NSP=NSP+2
      XMIN(I)=1.0E10
      XMAX(I)=-1.0E10
59    CONTINUE
C     SPECIFY INTERNAL TIME BREAKS
      CALL INTLET
      WRITE (KPTR,1310)
1310  FORMAT (22H INTERNAL TIME BREAKS ,17H INDEX  CODE  NDT)
```

```
      DO 65 I=1,NCODES
      IF (NSTEP(I) .EQ. 0) GO TO 65
      WRITE (KPTR,1320) I, ICODES(I), NSTEP(I)
1320  FORMAT (26X, I2, 3X, A4, 2X, I2)
65    CONTINUE

C     INITIALIZE SOME MORE QUANTITIES
      QXT(1)=-1.0
      QYT(1)=-1.0
      TCR(1,1)=9999.0
      NXP1=NX+1
      N=NX+NU
      TIME=T0
      PTL=TIME
      XL=TIME
      TNEXT=TIME
      ND2=0

C     MAIN COMPUTATIONAL LOOP BEGINS HERE
C     START BY HANDLING INTERNAL AND EXTERNAL BREAKS
70    CALL UPDATE(TNEXT)

C     IF EITHER THE A OR B MATRIX HAS CHANGED, DISCRETIZE THEM
C     THEN COMPUTE NEW FEEDBACK GAINS
100   IF (IFLAG(1)+IFLAG(2) .EQ. 0) GO TO 150
      CALL DSCRT(N,AA,DEL,A,WORK,4)
      CALL EQUATE(BD,WORK(1,NXP1),N,NU)
      CALL EQUATE(SAV,A(1,NXP1),N,NU)
150   CALL GPFBGN(WORK)

C     COMPUTE EXP(A*T) = (EXP(A*DEL))**NPRED
205   IF (IFLAG(1)+IFLAG(2)+IFLAG(10) .EQ. 0) GO TO 210
      CALL IDENT(N,EA,1.0)
      IF (NPRED.EQ.0) GO TO 210
      CALL EQUATE(EA,A,N,N)
      IF (NPRED.EQ.1) GO TO 210
      DO 208 I=2,NPRED
      CALL MMUL(A,EA,N,N,N,COM2)
      CALL EQUATE(EA,COM2,N,N)
208   CONTINUE

C     DO INCREMENTS TO MEANS
C     P DISTRIBUTES INCREMENT BETWEEN EST. AND ERR.
C     SO THAT XBAR-XHN=ENF
C     ALSO INCREMENT SIGMA MATRIX AND YBAR=DISPLAY MEANS
210   IF (IFLAG(22).EQ.0) GO TO 230
C     P=1.0
      P=0.0
      CALL VMAT1(EA,XMINC,N,N,YBAR)
      DO 216 I=1,N
      XHN(I)=XHN(I)+YBAR(I)*P
      ENF(I)=ENF(I)+XMINC(I)*(1.0-P)
      XBAR(I)=XBAR(I)+YBAR(I)
      DO 215 J=1,N
215   SIGMA(I,J)=SIGMA(I,J)+XMINC(I)*XMINC(J)*(1.0-P)**2
216   CONTINUE
      CALL VMAT1(C,XBAR,NY,N,YBAR)

C     DO INCREMENTS TO COVARIANCES
230   IF (IFLAG(23).EQ.0) GO TO 250
      P=0.62
      DO 240 I=1,N
      XCOV(I,I)=XCOV(I,I)+XSINC(I)**2
      ECOV(I,I)=ECOV(I,I)+XSINC(I)**2*(1.0-P)
      SIGMA(I,I)=SIGMA(I,I)+XSINC(I)**2*(1.0-P)
```

```
      DO 235 J=1,N
235   YCOV(I,J)=0.0
      YCOV(I,I)=P*XSINC(I)**2
240   CONTINUE
      CALL MMUL(EA,YCOV,N,N,N,COM1)
      CALL MAT6S(N,N,COM1,EA,XHCOV)
      CALL MAT3(NY,N,C,XCOV,YCOV)

C     COMPUTE MOTOR NOISE
250   DO 280 I=1,NU
      J=I+NX
C     CHANGE NEXT STATEMENT DEPENDING ON THE CONTROLLER
      VUO(I)=VU(I)+PU(I)*ABS(XCOV(J,J))
      WO(I+NW)=VUO(I)
280   CONTINUE

C     UPDATE YNOM FOR NEW XNOM
      IF (IFLAG(25).EQ.1) CALL VMAT1(C,XNOM,NY,N,YNOM)

C     COMPUTE OBSERVATION NOISE
      DO 335 I=1,NY
      YM=YBAR(I)-YNOM(I)
      YS=SQRT(YCOV(I,I))
      GTH(I)=XGAIN(TH(I),YM,YS)**2
      VYO(I)=VY(I)+PY(I)*(YM**2+YS**2)
      VYO(I)=VYO(I)/GTH(I)/ATTN(I)
335   CONTINUE

C     PROPAGATE COVARIANCES ONE TIME STEP AND INCREMENT TIME
      CALL COVAR(WORK)
320   TIME=TIME+DEL

C     ADJUST OUTPUTS FOR NOMINAL PATH
C     PROBLEM DEPENDENT
      CALL ADJNOM

C     DO OUTPUT FOR THIS TIME STEP
340   CALL INFORM(PTL,ND2,COM1,NSP)

C     IF THE TIME IS NOT EXPIRED DO ANOTHER ITERATION
      IF (TIME+0.0001 .LT. TEND+TEXT) GO TO 70

C     TIME IS EXPIRED.  DO OUTPUT AND START AGAIN
15    CONTINUE
      XH=TEND+TEXT
      CALL PRINTR(WORK,NSP)
      GO TO 1

C     RETURN IF AN EOF WAS READ
400   RETURN

      END
```

```
C       TIVCMP - COMPUTATIONS FOR TIVAR
C       INCLUDES
C         1 - SUBROUTINE GPFBGN
C         2 - SUBROUTINE TERMG
C         3 - SUBROUTINE COVAR
C         4 - SUBROUTINE FINAL


        SUBROUTINE GPFBGN(X)
C       GENERAL PURPOSE FEEDBACK GAIN COMPUTATION

        DIMENSION X(1), QXU(60)

        COMMON
     1  /COMMUN/ NCD, PINT, NSP, ICODES(32), IFLAG(32)
     2  /INOU/   KIN, KOUT, KPTR
     4  /MAIN1/  NDIM, NDIM1, COM1(1)
     5  /MAIN2/  Z(1)
     8  /COMP5/  ADCL(1)
     9  /INPTS/  BD(60), DUM(60), AD(1)
     A  /INPTX/  N, NX, NX1, NU, AC(1)
     B  /INPTY/  NY, C(1)
     C  /INPTW/  NW, E(1)
     E  /TIMES/  TIME, DEL
     F  /WEIGHT/ QX(30), QY(30), QR(30), PSS(1)
     J  /GAINBK/ CGN(1)

1       NM1=N-1
        NN=N*NDIM
        NNU=NU*NDIM
        NNW=NW*NDIM
        NNX=NX*NDIM+1
        CALL EQUATE(AD(NNX),DUM,N,NU)

C       SKIP STEADY STATE GAIN COMPUTATION UNLESS NEWG FLAG IS SET
        IF (IFLAG(21) .EQ. 0) GO TO 100

C       STEADY STATE GAIN COMPUTATION
2       DO 5 I=1,NY
        C1=QY(I)*DEL/2.0
        DO 5 J=I,NN,NDIM
        X(J)=C1*C(J)
5       CONTINUE
        CALL MAT2A(NY,N,C,X,X)
        II=1
        DO 6 I=1,N
        X(II)=X(II)+QX(I)*DEL/2.0
        II=II+NDIM1
6       CONTINUE
        CALL MAT3A(N,N,AD,X,Z)
        CALL MSCALE(QXU,X(NNX),N,NU,DEL)

        II=1
        DO 9 I=1,NU
        QR(I+NU)=DEL*QR(I)
        CALL VSCALE(COM1(II),QXU(II),N,-1.0/QR(I+NU))
        II=II+NDIM
9       CONTINUE
        DO 10 I=1,N
        DO 10 J=I,NN,NDIM
        ADCL(J)=AD(J)
        Z(J)=Z(J)+X(J)
        X(J)=AD(J)
10      CONTINUE
```

```
      CALL MAT5S(BD,COM1,N,NU,N,X)
      CALL MAT6S(N,NU,QXU,COM1,Z)
      CALL TRANS1(N,X,X)
      CALL KFLTR(N,NU,X,BD,Z,QR(NU+1),PSS,DUM,CGN)
      CALL MMUL(X,DUM,N,N,NU,Z)

      DO 15 I=1,NU
      II=(I-1)*NDIM
      C1=1.0/QR(NU+I)
      DO 15 J=I,NN,NDIM
      II=II+1
      CGN(J)=Z(II)+QXU(II)*C1
      COM1(J)=-CGN(J)
15    CONTINUE
      CALL MMULS(BD,COM1,N,NU,N,ADCL)

C     GET EQUIVALENT CONTINUOUS GAINS
      C1=-DEL
      DO 20 L=1,2
      CALL EQUATE(X,AC,N,N)
      CALL EQUATE(X(NX+1),COM1,NU,N)
      CALL DSCRT(N,X,DEL,Z,COM1,4)
      CALL GMINV(N,N,COM1,Z,MR,1)
      CALL MMUL(CGN,Z,NU,N,N,COM1)
      IF (L.EQ.2) C1=DEL
      CALL MSCALE(COM1,COM1,NU,N,C1)
20    CONTINUE

      WRITE (KPTR,1030)
1030  FORMAT (43H COMPUTED STEADY STATE DISCRETE CTRL GAINS )
      CALL MATPRT(NU,N,CGN,5HDGAIN)
      WRITE (KPTR,1032)
1032  FORMAT (35H EQUIVALENT CONTINUOUS GAINS LX,LU )
      CALL MATPRT(NU,N,COM1,5HCGAIN)
      CALL GMINV(NU,NU,COM1(NNX),Z(NNX),MR,1)
      CALL MMUL(Z(NNX),COM1,NU,NU,NX,Z)
      WRITE (KPTR,1033)
1033  FORMAT (31H EQUIVALENT CONTINUOUS LOPT,TN )
      CALL MATPRT(NU,N,Z,5HL*,TN)

C     IF GAINS WERE NOT READ IN, COMPUTE TERMINAL GAINS
100   IF (IFLAG(19) .EQ. 1) GO TO 150
      IF (IFLAG(18) .EQ. 1) GO TO 170
      CALL TERMG(X,QXU)
      GO TO 200

C     DGAINS WERE READ IN
150   CALL EQUATE(COM1,AC,NX,N)
      CALL MSCALE(COM1(NX+1),CGN,NU,N,-1.0)
      CALL DSCRT(N,COM1,DEL,CGN,Z,4)
      CALL MMUL(COM1(NX+1),Z,NU,N,N,CGN)
      CALL MSCALE(CGN,CGN,NU,N,-1.0/DEL)
      WRITE (KPTR,1152)
1152  FORMAT (37H EQUIVALENT DISCRETE GAINS GENERATED )
      CALL MATPRT(NU,N,CGN,5HDGAIN)
170   CONTINUE
C     KILL TCOST FLAG

200   CALL MMUL(BD,CGN(NNX),N,NU,NU,E(NNW+1))
      DO 205 I=NNX,NN,NDIM
      II=I+NM1
      DO 205 J=I,II
      K=J-NX*NDIM
      L=K+NNW
      DUM(K)=AD(J)
      AD(J)=AD(J)-E(L)
```

```
      E(L)=E(L)/DEL
205   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TERMG(X,QXU)
C     COMPUTES CONTROL GAINS BASED ON TERMINAL COSTS (FORWARD TIME)

      DIMENSION
     1 X(1), QXU(1), BR(60), FM(225), PHN(225)

      COMMON
     1 /COMMUN/ NCD, PINT, NSP, ICODES(32), IFLAG(32)
     2 /INOU/   KIN, KOUT, KPTR
     4 /MAIN1/  NDIM, NDIM1, COM1(1)
     5 /MAIN2/  COM2(1)
     8 /COMP5/  ADCL(1)
     9 /INPTS/  BD(60), SAV(60), AD(1)
     A /INPTX/  N, NX, NX1, NU
     B /INPTY/  NY, C(1)
     E /TIMES/  TIME, DEL, TO, TEND, TEXT, TD, NPRED
     F /WEIGHT/ QX(30), QY(30), QR(30), PSS(1)
     G /TCOST/  QXT(30), QYT(30), PINV(1)
     J /GAINBK/ CGN(1)

1     NN=N*NDIM
C     RETURN IF NO TERMINAL COSTS
      IF (QXT(1)+QYT(1) .LE. -1.5) RETURN
C     CHECK TIME TO GO
      NGO=(TEND-TIME+0.0001)/DEL-1
      IF (NGO) 150,50,5

C     DO THIS PART IF NGO IS POSITIVE
C     AND IF QXT OR QYT HAVE CHANGED
5     IF (IFLAG(27)+IFLAG(28) .EQ. 0) GO TO 50
      IF (TIME .GT. TO+0.0001) GO TO 8
      NGO=NGO-NPRED
8     CALL EQUATE(X,ADCL,N,N)
      CALL MMUL(PSS,BD,N,N,NU,COM1)
      CALL MAT2A(N,NU,BD,COM1,COM1)

      II=1
      DO 10 I=1,NU
      COM1(II)=COM1(II)+QR(I)*DEL
      II=II+NDIM1
10    CONTINUE

      CALL GMINV(NU,NU,COM1,COM2,MR,1)
      CALL MMUL(BD,COM2,N,NU,NU,BR)
      CALL MAT6(N,NU,BR,BD,PHN)
      CALL DINTEG(N,X,PHN,PINV,NGO)
      TPT=TIME

      C1=0.0
      DO 15 I=1,NY
      IF (QYT(1) .GT. -0.5) C1=QYT(I)
      DO 15 J=I,NN,NDIM
      X(J)=C1*C(J)
15    CONTINUE
      CALL MAT2A(NY,N,C,X,X)

      II=1
      DO 17 I=1,N
      DO 16 J=I,NN,NDIM
      X(J)=X(J)-PSS(J)
16    CONTINUE
      IF (QXT(1) .GT. -0.5) X(II)=X(II)+QXT(I)
      II=II+NDIM1
17    CONTINUE
C     WRITE (KOUT,1800) TEND
C1800 FORMAT (27H TERMINAL CONDITION AT TIME,F10.3)
```

```
C       CALL MATIO(X,N,N,3)
        CALL GMINV(N,N,X,FM,MR,1)
        DO 25 I=1,N
        DO 25 J=I,NN,NDIM
        COM2(J)=FM(J)+PINV(J)
25      CONTINUE
        CALL GMINV(N,N,COM2,COM1,MR,1)
        CALL MMUL(COM1,PHN,N,N,N,COM2)
        CALL MAT2A(N,N,PHN,COM2,COM1)
        CALL MSCALE(BR,BR,N,NU,-1.0)
        GO TO 100

C       UPDATE PINVERSE
C       DO THIS PART IF NGO IS ZERO
C       OR IF NEITHER QXT OR QYT HAVE CHANGED
50      IF (TIME-TSS) 62,62,60
60      CALL MMUL(ADCL,FM,N,N,N,COM1)
        CALL MAT2(N,NU,BR,BD,FM)
        CALL MAT6S(N,N,COM1,ADCL,FM)

62      DO 70 I=1,N
        DO 70 J=I,NN,NDIM
        COM2(J)=FM(J)+PINV(J)
70      CONTINUE
        CALL GMINV(N,N,COM2,COM1,MR,1)
        CALL MMUL(COM1,PHN,N,N,N,COM2)
        CALL MAT2A(N,N,PHN,COM2,COM1)

C       GET NEW CONTROL GAINS
C       DO THIS PART UNLESS NGO IS NEGATIVE
100     DO 110 I=1,N
        DO 110 J=I,NN,NDIM
        X(J)=PSS(J)+COM1(J)
110     CONTINUE
C       WRITE (KOUT,1900) TIME
C1900   FORMAT (26H DIFFERENCE MATRIX AT TIME,F10.3)
C       CALL MATIO(COM1,N,N,3)

        CALL MMUL(X,BD,N,N,NU,COM1)
        CALL MAT2A(N,NU,BD,COM1,COM2)
        II=1
        DO 112 I=1,NU
        COM2(II)=COM2(II)+QR(I)*DEL
        II=II+NDIM1
112     CONTINUE

        CALL GMINV(NU,NU,COM2,X,MR,1)
        CALL TRANS2(N,NU,QXU,COM2)
        CALL MAT5AS(COM1,AD,NU,N,N,COM2)
        CALL MMUL(X,COM2,NU,NU,N,CGN)
        IF (TIME.LT.TPT) GO TO 150
115     TPT=TIME+PINT
        WRITE (KOUT,1950) TIME
1950    FORMAT (21H CONTROL GAIN AT TIME,F10.3)
        CALL MATIO(CGN,NU,N,3)

150     RETURN

        END
```

```
      SUBROUTINE COVAR(X5)
C     PROPAGATES COVARIANCES ONE TIME STEP

      DIMENSION
     1 X5(1), DELF(5,30)

      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1, X1(1)
     5 /MAIN2/  X3(1)
     9 /INPTS/  BD(60), SAV(60), A(1)
     A /INPTX/  N, NX, NX1, NU
     B /INPTY/  NY, C(1)
     C /INPTW/  NW, E(1)
     D /INPTWD/ NWD, F(1)
     E /TIMES/  TIME, DEL, TO, TEND, TEXT, TD, NPRED, EA(1)
     H /NOISE/  VU(30), VY(30), WO(30), SIGMA(1)
     J /GAINBK/ CGN(1)
     L /COVX/   XBAR(30), XCOV(1)
     M /COVY/   YBAR(30), YCOV(1)
     N /COVXHE/ DUM(30), XHECOV(1)
     O /COVXH/  XHN(30), XHCOV(1)
     P /COVEF/  ENF(30), ENP(30), ECOV(1)
     Q /NOMNL/  NTF, XNOM(1)

      IF (TIME .GT. TO+0.001) GO TO 8

C     INITIALIZATION
      NWU=NW+NU
      NM1=N-1
      NN=N*NDIM
      DO 1 K=1,NWD
      DELF(K,1)=FDOT(K,TIME)*DEL
1     CONTINUE
      TCOR=1.0
      TCOR=SQRT(TCOR/DEL)
      IF (NPRED.EQ.0) GO TO 8

C     INITIALIZE FOR THE TIME DELAY
      DO 2 I=1,NPRED
      CALL FINAL(X5)
      IFLAG(26)=0
      TIME=TIME+DEL
      DO 2 K=1,NWD
      DELF(K,I+1)=FDOT(K,TIME)*DEL
2     CONTINUE

C     INTEGRATE THE DRIVING AND MOTOR NOISES
8     II=1
      DO 9 I=1,NWU
      TMP=WO(I)*DEL
      CALL VSCALE(X5(II),E(II),N,TMP)
      II=II+NDIM
9     CONTINUE
      CALL MAT2(N,NWU,E,X5,X5)

C     COMPUTE COVARIANCES AND ESTIMATOR GAIN
      DO 10 I=1,NY
      TMP=10000.0
      IF (VY(I) .GE. 1.0E-5) TMP=DEL/VY(I)
      DO 10 J=I,NN,NDIM
      X3(J)=C(J)*TMP
10    CONTINUE
```

```
      CALL MAT2A(NY,N,C,X3,X3)
      CALL MMUL(SIGMA,X3,N,N,N,XCOV)

      DO 11 I=1,NN,NDIM1
      XCOV(I)=XCOV(I)+1.0
11    CONTINUE

      CALL GMINV(N,N,XCOV,YCOV,MR,1)
      CALL MMUL(YCOV,SIGMA,N,N,N,XCOV)
      CALL MMUL(XCOV,X3,N,N,N,X1)
      CALL DIAG2(N,YCOV,X1,-1.0,1.0)
      CALL MMUL(EA,X1,N,N,N,X3)
      CALL MAT2(N,N,XCOV,X1,X1)
      CALL MMUL(A,X1,N,N,N,XCOV)
      CALL MAT6S(N,N,XCOV,A,X5)
      CALL MMUL(A,YCOV,N,N,N,X1)
      CALL MMUL(EA,SIGMA,N,N,N,YCOV)
      CALL MMUL(X1,ECOV,N,N,N,XCOV)
      CALL MAT6S(N,N,XCOV,X1,X5)
      CALL VMAT2(XHN,X3,ENF,N,N,DUM)
      CALL VMAT1(X1,ENF,N,N,XBAR)

      DO 15 I=1,N
      ENF(I)=XBAR(I)
      DO 15 J=I,NN,NDIM
      SIGMA(J)=ECOV(J)-SIGMA(J)
      ECOV(J)=X5(J)
15    CONTINUE

      CALL MMUL(X3,SIGMA,N,N,N,X5)
      NNX=NX*NDIM
      DO 20 I=1,N
      IM1=I-1
      DO 20 J=I,NN,NDIM
      XCOV(J)=XHECOV(J)+(YCOV(J)+X5(J))/2.0
      YCOV(J)=A(J)
      IF (J .GT. NNX) GO TO 20
      YCOV(J)=YCOV(J)-DOT3(NU,BD(I),CGN(J-IM1))
20    XHECOV(J)=XHECOV(J)+X5(J)

      CALL MMUL(YCOV,XHECOV,N,N,N,X5)
      CALL MAT5(X5,X1,N,N,N,XHECOV)
      CALL MAT5(XCOV,X3,N,N,N,X5)
      CALL MMUL(X1,SIGMA,N,N,N,X3)
      CALL MAT2(N,N,X3,X1,SIGMA)
      CALL MMUL(EA,ECOV,N,N,N,X3)
      CALL VMAT1(YCOV,DUM,N,N,XHN)

      DO 30 II=1,NN,NDIM1
      K=(II+NDIM)/NDIM1
      DUM(K)=0.0
      I=II
      DO 30 J=II,NN,NDIM
      XHCOV(J)=XHCOV(J)+X5(J)+X5(I)
      SIGMA(J)=ECOV(J)-SIGMA(J)
      X1(J)=XHECOV(J)+X3(J)/2.0
      X1(I)=XHECOV(I)+X3(I)/2.0
      SIGMA(I)=SIGMA(J)
      XHCOV(I)=XHCOV(J)
      I=I+1
30    CONTINUE

      CALL MMUL(YCOV,XHCOV,N,N,N,X3)
      CALL MAT2(N,N,X3,YCOV,XHCOV)
      CALL MAT5(X1,EA,N,N,N,X3)
      DO 40 II=1,NN,NDIM1
```

```
      I=II
      DO 40 J=II,NN,NDIM
      XCOV(J)=XHCOV(J)+X3(J)+X3(I)
      XCOV(I)=XCOV(J)
      I=I+1
40    CONTINUE

      II=1
      DO 50 I=1,NWD
      TMP=TCOR*DELF(I,1)
      CALL VSCALE(X5(II),F(II),NX,TMP)
      II=II+NDIM
50    CONTINUE

      CALL VMAT2(ENF,F,DELF,NX,NWD,ENF)
      CALL MAT6S(NX,NWD,X5,X5,SIGMA)
      IF (NPRED .EQ. 0) GO TO 70
      II=1
      DO 51 I=1,NWU
      TMP=SQRT(WO(I)*DEL)
      CALL VSCALE(X1(II),E(II),N,TMP)
      II=II+NDIM
51    CONTINUE

      DO 60 I=1,NPRED
      CALL MAT6S(N,NWU,X1,X1,XCOV)
      IF (I .EQ. NPRED) GO TO 53
      CALL MMUL(A,X1,N,N,NWU,X3)
      CALL EQUATE(X1,X3,N,NWU)
53    DO 55 K=1,NWD
      DELF(K,I)=DELF(K,I+1)
55    CONTINUE
      IF (I .GT. 1) CALL VMAT1(A,ENP,NX,NX,DUM)
      CALL VMAT2(DUM,F,DELF(1,I),NX,NWD,ENP)
60    CONTINUE
70    CALL VMAT2(ENP,EA,ENF,N,N,DUM)

      CALL FINAL(X5)
      DO 75 I=1,N
      XBAR(I)=XHN(I)+DUM(I)+XNOM(I)
75    CONTINUE
      CALL MAT3(NY,N,C,XCOV,YCOV)
      CALL VMAT1(C,XBAR,NY,N,YBAR)

      DO 80 K=1,NWD
      DELF(K,NPRED+1)=FDOT(K,TIME+DEL)*DEL
80    CONTINUE

      RETURN
      END
```

```
      SUBROUTINE FINAL(X)
C     COMPUTES NOMINAL PATH BASED ON TERMINAL CONDITION
C     CONSTRAINED BY MINIMIZING CTRL RATE

      DIMENSION
     1 QO(30), X(1)

      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32)
     4 /MAIN1/  NDIM, NDIM1, COM1(1)
     5 /MAIN2/  Y(1)
     6 /COMP1/  ADI(1)
     7 /COMP3/  Z(1)
     9 /INPTS/  BD(60), SAV(60), AD(1)
     A /INPTX/  N, NX, NX1, NU, AA(1)
     B /INPTY/  NY, C(1)
     E /TIMES/  TIME, DEL, TO, TEND, TEXT, TD, NPRED
     F /WEIGHT/ QX(30), QY(30), QR(30)
     Q /NOMNL/  NT, XNOM(30), YNOM(30), P(1)

C     P(1)=9999 IS A FLAG TO AVOID COMPUTATION
      IF (P(1) .EQ. 9999.0) RETURN
      NN=N*NDIM
      NGO=(TEND-TIME+0.0001)/DEL
      IF (NGO .LE. NT) GO TO 50
C     ASSUMES SYSTEM IS OUTPUT CONTROLLABLE

      IF (IFLAG(26).EQ.0) GO TO 50
C     A NEW TCR MATRIX WAS READ IN
      NNX=NX*NDIM
      II=1
      DO 6 I=1,NU
      C1=1.0/(QR(I)*DEL)
      L=II+N-1
      DO 5 J=II,L
      COM1(J)=BD(J)*C1
      AD(J+NNX)=SAV(J)
5     CONTINUE
      II=II+NDIM
6     CONTINUE

      CALL MAT2(N,NU,COM1,BD,Z)
      CALL EQUATE(ADI,AD,N,N)
      CALL DINTEG(N,ADI,Z,X,NGO)
      CALL MAT3(NT,N,P,X,COM1)
      CALL GMINV(NT,NT,COM1,X,MR,1)
      CALL MMUL(P,Z,NT,N,N,COM1)
      CALL VMAT2(P(NN+1),COM1,XNOM,NT,N,QO)
      CALL VMAT1(X,QO,NT,NT,YNOM)
      CALL TRANS2(NT,N,COM1,Y)
      CALL VMAT1(Y,YNOM,N,NT,QO)
      CALL TRANS1(N,AD,COM1)
      CALL GMINV(N,N,COM1,ADI,MR,1)
      CALL EQUATE(AD(NNX+1),SAV,N,NU)

C     UPDATE XNOM AND YNOM
50    IF (NGO .GT. 0) GO TO 75
      DO 70 I=1,N
      QO(I)=0.0
70    CONTINUE

75    II=1
      CALL VMAT1(ADI,QO,N,N,YNOM)
      DO 77 I=1,NU
      COM1(I)=-DOT(N,BD(II),YNOM)/(QR(I)*DEL)
      II=II+NDIM
```

```
77      CONTINUE

        CALL VMAT1(AD,XNOM,N,NX,Q0)
        CALL VMAT2(Q0,SAV,XNOM(NX+1),N,NU,Q0)
        CALL VMAT2(Q0,BD,COM1,N,NU,XNOM)
        CALL VSCALE(Q0,YNOM,N,1.0)
        CALL VMAT1(C,XNOM,NY,N,YNOM)

        RETURN
        END
```

```
C       TIVIO - I/O FOR TIVAR
C       INCLUDES
C         1 - SUBROUTINE INTLET
C         2 - SUBROUTINE UPDATE
C         3 - SUBROUTINE OUTLET
C         4 - SUBROUTINE INFORM
C         5 - SUBROUTINE PRINTR

        SUBROUTINE INTLET
C       SPECIFY INTERNAL UPDATES
C       NSTEP(I) REFERS TO THE ITH TYPE OF UPDATE
C       NSTEP(I)=0  NO INTERNAL UPDATE
C       NSTEP(I)=NDT  NDT DELS BETWEEN SUCCESSIVE INTERNAL UPDATES

        COMMON
     1  /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1           NSTEP(32), LPRNT(60), XMAX(60), XMIN(60)
     2  /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK

C       READ NSTEP(I) FROM SEVERAL CARDS
100     READ (KIN,1100) (NSTEP(I), I=1,NCODES)
1100    FORMAT (16I5)
        RETURN
        END
```

```
      SUBROUTINE UPDATE(TNEXT)
C     PERFORMS EXTERNAL UPDATES

      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1          NSTEP(32)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1
     A /INPTX/  N, NX, NX1, NU, AA(1)
     B /INPTY/  NY, C(1)
     C /INPTW/  NW, E(1)
     D /INPTWD/ NWD, F(1)
     E /TIMES/  TIME, DEL, T0, TEND, TEXT, TD, NPRED
     F /WEIGHT/ QX(30), QY(30), QR(30)
     G /TCOST/  QXT(30), QYT(30)
     H /NOISE/  VU0(30), VY0(30), W0(30)
     I /RATIOS/ PU(30), VU(30), PY(30), VY(30), TH(30), GTH(30)
     J /GAINBK/ GAINS(1)
     K /INCRE/  XMINC(30), XSINC(30), ATTN(30)
     Q /NOMNL/  NTF, XNOM(30), YNOM(30), TCR(1)

      DATA
     1 LEND, LZ, LTIME, PI /3HEND, 1HZ, 4HTIME, 3.14159/

      NN=N+1
      NNX=NX+1
      NNN=NX*NDIM+1
      TMAX=TEND+TEXT+0.1
      TDMAX=29*DEL

C     TAKE CARE OF THE INTERNAL UPDATES
      DO 100 I=1,NCODES
      IFLAG(I)=0
      IF (NSTEP(I).EQ.0) GO TO 100
      TTME=TIME-T0
      IF (MOD(INT(TTME/DEL+0.1),NSTEP(I)) .EQ. 0) CALL INTNEW(I)
100   CONTINUE

C     RETURN IF NOT TIME FOR THE NEXT EXTERNAL UPDATE
110   IF (TIME+0.001 .LT. TNEXT) GO TO 220

C     SPECIFY THE TYPE OF EXTERNAL UPDATE (AND TIME IF LTIME)
120   READ (KIN,1030) IDEN, BRKT
1030  FORMAT (A5,5X,E10.0)
      IF (EOF(KIN)) 135,130

C     CHECK FOR LEND
130   IF (IDEN.NE.LEND) GO TO 140
135   TNEXT=TMAX
      GO TO 110

C     CHECK FOR LTIME
140   IF (IDEN.NE.LTIME) GO TO 150
      TNEXT=BRKT
      GO TO 110

C     CHECK FOR LZ (NULL UPDATE) - FOR HRA COMPATABILITY
150   IF (IDEN.EQ.LZ) GO TO 120

C     SEARCH THROUGH THE UPDATE CODES, ICODE(KEY)
      DO 160 KEY=1,NCODES
      IF (IDEN.EQ.ICODES(KEY)) GO TO 170
160   CONTINUE

C     CODE WAS ILLEGAL
      WRITE (KOUT,1165) IDEN
```

```
1165     FORMAT (1H ,5HCODE ,A5,11H IS ILLEGAL)
         CALL EXIT

C        DO THE SPECIFIED EXTERNAL UPDATE
170      IFLAG(KEY)=1
         IO=2
         WRITE (KPTR,1175) TIME, IDEN
1175     FORMAT (/,25H EXTERNAL UPDATE AT TIME ,F8.3,4X,4HCODE,2X,A5)
         GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     1          21,22,23,24,25,26,27,28,29,30), KEY

C        SYSTEM DYNAMICS - A, B, C, D, E
1        CALL MATIO(AA,NX,NX,IO)
         GO TO 120
2        CALL MATIO(AA(NNN),NX,NU,IO)
         GO TO 120
3        CALL MATIO(C,NY,NX,IO)
         GO TO 120
4        CALL MATIO(C(NNN),NY,NU,IO)
         GO TO 120
5        CALL MATIO(E,NX,NW,IO)
         GO TO 120

C        COST FUNCTIONAL - QX, QY, QU, QR
6        CALL VECTIO(QX,NX,IO)
         GO TO 120
7        CALL VECTIO(QY,NY,IO)
         GO TO 120
8        CALL VECTIO(QX(NNX),NU,IO)
         GO TO 120
9        CALL VECTIO(QR,NU,IO)
         GO TO 120

C        TIME DELAY - TD
10       CALL VECTIO(TD,1,1)
         NPRED=IFIX((TD+0.0001)/DEL)
         TD=DEL*NPRED
         CALL VECTIO(TD,1,3)
         GO TO 120

C        NOISES - PN, MNA, MNR, SNA, SNR
11       CALL VECTIO(WO,NW,IO)
         GO TO 120
12       CALL VECTIO(VU,NU,IO)
         GO TO 120
13       CALL VECTIO(PU,NU,IO)
         DO 200 I=1,NU
200      PU(I)=PI*10.0**(PU(I)/10.0)
         GO TO 120
14       CALL VECTIO(VY,NY,IO)
         GO TO 120
15       CALL VECTIO(PY,NY,IO)
         DO 201 I=1,NY
201      PY(I)=PI*10.0**(PY(I)/10.0)
         GO TO 120

C        THRESHOLDS AND ATTENTION - TH, ATTN
16       CALL VECTIO(TH,NY,IO)
         GO TO 120
17       CALL VECTIO(ATTN,NY,IO)
         GO TO 120

C        CONTINUOUS AND DISCRETE CTRL GAINS - CGAIN, DGAIN
18       CALL MATIO(GAINS,NU,N,IO)
         GO TO 120
```

```
19      CALL MATIO(GAINS,NU,N,IO)
        GO TO 120

C       PRINT INTERVAL - PNTVL
20      CALL VECTIO(PNTVL,1,IO)
        GO TO 120

C       COMPUTE NEW GAINS - NEWG
21      GO TO 120

C       INCREMENTS TO STATES - XMINC, XSINC
22      CALL VECTIO(XMINC,N,IO)
        GO TO 120
23      CALL VECTIO(XSINC,N,IO)
        GO TO 120

C       DETERMINISTIC INPUT (F MATRIX) - F
24      CALL MATIO(F,NX,NWD,IO)
        GO TO 120

C       NOMINAL PATH VECTOR - XNOM
25      CALL VECTIO(XNOM,N,IO)
        GO TO 120

C       TERMINAL CONDITION MATRIX - TCR
26      CALL MATIO(TCR,NTF,NN,IO)
        GO TO 120

C       CALL AN INTERNAL UPDATE - INT
27      READ (KIN,1210) KEY
1210    FORMAT (I2)
        WRITE (KOUT,1220) KEY
1220    FORMAT (17H INTNEW PART NO. ,I3)
        CALL INTNEW(KEY)
        GO TO 120

C       TERMINAL COSTS - QXT, QYT
28      CALL VECTIO(QXT,NX,IO)
        GO TO 120
29      CALL VECTIO(QYT,NY,IO)
        GO TO 120

C       DUMMY UPDATE - DUMMY
30      CONTINUE
        GO TO 120

C       NO MORE EXTERNAL UPDATES AT THIS TIME
220     CONTINUE

C       GAIN COMPUTATION REQUEST IS CANCELLED BY READ-INS
        IF (IFLAG(18).EQ.1 .OR. IFLAG(19).EQ.1) IFLAG(21)=0
        RETURN

        END
```

```
      SUBROUTINE OUTLET
C     SPECIFY VARIABLES FOR OUTPUT
C     LPRNT(1,20) STATES; LPRNT(21-40) OUTPUTS; LPRNT(41-60) CONTROLS
C     LPRNT(I)=1 PRINT; LPRNT(I)=2 PRINT & PLOT; LPRNT(I)=3 PLOT

      COMMON
    1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
    1          NSTEP(32), LPRNT(60)
    2 /INOU/   KIN, KOUT, KPTR
    4 /MAIN1/  NDIM

C     READ LPRNT(I) FROM ONE CARD
100   READ (KIN,1100) (LPRNT(I), I=1,60)
1100  FORMAT (60I1)
      RETURN

      END
```

```
      SUBROUTINE INFORM(PTL,NPTL,BUF,KDIM)
C     DO OUTPUT FOR A SINGLE TIME STEP
      DIMENSION BUF(KDIM)
      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1          NSTEP(32), LPRNTX(20), LPRNTY(20), LPRNTU(20),
     1          XMAX(20), YMAX(20), UMAX(20),
     1          XMIN(20), YMIN(20), UMIN(20)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1
     A /INPTX/  N, NX
     E /TIMES/  TIME
     L /COVX/   XBAR(30), XCOV(1)
     M /COVY/   YBAR(30), YCOV(1)
      LL=1
C     CHECK IF TIME FOR A PRINTOUT
      IF (TIME-PTL+0.001 .LT. PINTVL) RETURN
C     DO A PRINTOUT
      PTL=TIME
      NPTL=NPTL+1
      BUF(1)=TIME
C     DO STATES
      DO 10 L=2,4,2
      DO 10 I=1,20
      LPI=LPRNTX(I)
      IF (LPI.NE.L .AND. LPI.NE.L-1) GO TO 10
      LL=LL+2
      BUF(LL-1)=XBAR(I)
      II=I*NDIM1-NDIM
      BUF(LL)=SQRT(ABS(XCOV(II)))
      IF (LPI.EQ.1) GO TO 10
      C1=XBAR(I)+BUF(LL)
      C2=XBAR(I)-BUF(LL)
      IF (C1.GE.XMAX(I)) XMAX(I)=C1
      IF (C2.LE.XMIN(I)) XMIN(I)=C2
10    CONTINUE
C     DO OUTPUTS
      DO 20 L=2,4,2
      DO 20 I=1,20
      LPI=LPRNTY(I)
      IF (LPI.NE.L .AND. LPI.NE.L-1) GO TO 20
      LL=LL+2
      BUF(LL-1)=YBAR(I)
      II=I*NDIM1-NDIM
      BUF(LL)=SQRT(ABS(YCOV(II)))
      IF (LPI.EQ.1) GO TO 20
      C1=YBAR(I)+BUF(LL)
      C2=YBAR(I)-BUF(LL)
      IF (C1.GE.YMAX(I)) YMAX(I)=C1
      IF (C2.LE.YMIN(I)) YMIN(I)=C2
20    CONTINUE
C     DO CONTROLS
      DO 30 L=2,4,2
      DO 30 I=1,20
      LPI=LPRNTU(I)
      IF (LPI.NE.L .AND. LPI.NE.L-1) GO TO 30
      J=I+NX
      LL=LL+2
      BUF(LL-1)=XBAR(J)
```

```
      JJ=J*NDIM1-NDIM
      BUF(LL)=SQRT(ABS(XCOV(JJ)))
      IF (LPI.EQ.1) GO TO 30
      C1=XBAR(J)+BUF(LL)
      C2=XBAR(J)-BUF(LL)
      IF (C1.GE.UMAX(I)) UMAX(I)=C1
      IF (C2.LE.UMIN(I)) UMIN(I)=C2
30    CONTINUE

C     WRITE THE BUFFER ON THE DISK AND RETURN
      WRITE (KDISK) BUF
      RETURN
      END
```

```
      SUBROUTINE PRINTR(BUF,KDIM)
C     DO ALL THE OUTPUT AT THE END OF A RUN

      DIMENSION
     1 BUF(KDIM), TITLE(3), LET(10,3)

      COMMON
     1 /COMMUN/ NCODES, PINTVL, NSP, ICODES(32), IFLAG(32),
     1          NSTEP(32), LPRNT(20,3), XMAX(60), XMIN(60)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     3 /PLOT1/  NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     3          NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT
     4 /MAIN1/  NDIM, NDIM1, GRAPH(1)
     A /INPTX/  STORE(1)

      DATA
     1  PS1, PS2,  LMEAN,       LSD
     1 /1HM, 1HS, 4HMEAN, 8HSTD  DEV/,
     2 (TITLE(I), I=1,3)
     2 /8H  STATE ,8H OUTPUT ,8HCONTROL /

      IBEG=ND2+1
      REWIND KDISK

      DO 10 I=1,ND2
      READ (KDISK) BUF
      II=I
      DO 9 L=1,KDIM
      STORE(II)=BUF(L)
9     II=II+ND2
10    CONTINUE

      DO 90 I=1,3
      M=0

      DO 30 L=1,20
      IQ=LPRNT(L,I)
      IF (IQ.EQ.0 .OR. IQ.EQ.3) GO TO 30
      M=M+1
      LET(M,I)=L
30    CONTINUE
      IF (M.EQ.0) GO TO 50

C     PRINT OPTIONS
      CALL PAGEFD(KPTR,1)
      WRITE (KPTR,35) (TITLE(I), LET(J,I), J=1,M)
35    FORMAT (1H ,9X,5(5X,A8,I4,5X))
      WRITE (KPTR,36) (LMEAN,LSD,J=1,M)
36    FORMAT (1H ,2X,4HTIME,2X,5(5X,A4,2X,A8,3X))
      LIM1=IBEG
      LIM2=LIM1+(2*M-1)*ND2

      DO 40 L=1,ND2
      WRITE (KPTR,45) STORE(L), (STORE(J),J=LIM1,LIM2,ND2)
45    FORMAT (1H ,F6.2,10(2X,1PE9.2))
      LIM1=LIM1+1
      LIM2=LIM2+1
40    CONTINUE

C     PLOT OPTIONS
50    IM1=(I-1)*20
      DO 90 K=2,4,2
      DO 90 L=1,20
      IQ=LPRNT(L,I)
      IF (IQ.NE.K .AND. IQ.NE.K-1) GO TO 90
      IF (IQ.EQ.1) GO TO 85
```

```
      YH=XMAX(L+IM1)
      YL=XMIN(L+IM1)
      IST=10
      IEND=IBEG+ND2-1
      PSYM=PS1
      CALL KPLOT(GRAPH,STORE,STORE(IBEG),0,0,0,0,0)

      DO 80 J=IBEG,IEND
      C1=STORE(J)
      C2=STORE(J+ND2)
      STORE(J)=C1-C2
      STORE(J+ND2)=C1+C2
80    CONTINUE

      IST=0
      PSYM=PS2
      CALL KPLOT(GRAPH,STORE,STORE(IBEG),0,0,0,0,0)
      IST=1
      CALL KPLOT(GRAPH,STORE,STORE(IBEG+ND2),0,0,0,0,0)
      WRITE (KPTR,82) TITLE(I), L
82    FORMAT (/,1H ,A8,I3)

85    IBEG=IBEG+2*ND2
90    CONTINUE

      RETURN
      END
```

```
      SUBROUTINE KPLOT(W,X,Y,NTAPE,IX,IY,NVAR,Y1)
      COMMON /PLOT1/
     1 NV,NH,NCPW,LW,VLH(4),NXES,NDIR,IST,NGLV,NGLH,BSYM,GSYM,
     2 PSYM,NDIM1,NDIM2,NO
      DIMENSION W(1),X(1),Y(1),Y1(NVAR),STORE(70),Q(4),IPX(4),K(3)
      EQUIVALENCE (Q(1),XL1),(Q(2),XH1),(Q(3),YL1),(Q(4),YH1)
      EQUIVALENCE (ISC,K(1)),(JSC,K(3))
      DATA IPX/3,4,1,2/
      IF (NH.GT.121) NH=121
C     NCPW IS THE NUMBER OF CHARACTERS PER WORD
C     (60 BIT WORD 6 BIT DISPLAY CODE ON CDC)
      NCPW=10
      LW=NH/NCPW+1
      IF ((IST/10).GT.0) NCOUNT=0
      NCOUNT=NCOUNT+1
      IF (NCOUNT.EQ.10) IST=1
      L=1
      DO 10 I=1,4
      Q(I)=-1.0E08*(-1)**I
      K(L)=1
      IF (VLH(L).EQ.VLH(L+1)) GO TO 10
      K(L)=0
      Q(I)=VLH(I)
10    IF (I.EQ.2) L=3
      IF (NTAPE.EQ.0) GO TO 1200
C     SKIP THIS PART IF PLOTTING FROM CORE
      IFLAG=0
      GO TO 40
1600  IFLAG=1
40    NN=0
      REWIND NTAPE
50    READ (NTAPE) Y1
C     GO TO 2800 ON EOF
      IF (EOF(NTAPE))2800,100
100   NN=NN+1
      IF (NN.LT.NDIM1) GO TO 50
      IF (IFLAG.EQ.1) GO TO 1700
      IF (ISC+JSC.EQ.0) GO TO 1710
300   NN=NN+1
      IF (ISC.EQ.0) GO TO 600
      XL1=AMIN1(XL1,Y1(IX))
      XH1=AMAX1(XH1,Y1(IX))
      IF (JSC.EQ.0) GO TO 200
600   YL1=AMIN1(YL1,Y1(IY))
      YH1=AMAX1(YH1,Y1(IY))
200   READ (NTAPE) Y1
      IF (EOF(NTAPE))1600,210
210   IF (NN-NDIM2) 300,300,1600
C     RESUME HERE
1200  IF (ISC.EQ.0) GO TO 1400
      DO 1300 I=NDIM1,NDIM2
      XL1=AMIN1(XL1,X(I))
1300  XH1=AMAX1(XH1,X(I))
1400  IF (JSC.EQ.0) GO TO 1700
      DO 1500 I=NDIM1,NDIM2
      YL1=AMIN1(YL1,Y(I))
1500  YH1=AMAX1(YH1,Y(I))
1700  IF (ISC.EQ.1) CALL ADJUST(XH1,XL1)
      IF (JSC.EQ.1) CALL ADJUST(YH1,YL1)
```

```
1710   IF (NDIR/10) 1720,1740,1720
1720   TMP=XL1
       XL1=XH1
       XH1=TMP
1740   IF (NDIR-10*(NDIR/10)) 1760,1780,1760
1760   TMP=YL1
       YL1=YH1
       YH1=TMP
1780   J=7*(NCOUNT-1)+1
       IF (J.EQ.1) CALL QINIT(W)
       STORE(J)=PSYM
       DO 1800 I=1,4
       IF (NXES.EQ.0) L=I+J
       IF (NXES.GT.0) L=IPX(I)+J
1800   STORE(L)=Q(I)
       STORE(J+5)=(NH-1)/(STORE(J+2)-STORE(J+1))
       STORE(J+6)=(NV-1)/(STORE(J+4)-STORE(J+3))
2200   IF (NTAPE.EQ.0) GO TO 2500

C      SKIP THIS PART IF PLOTTING FROM CORE
       DO 2400 I=NDIM1,NDIM2
       IF (NXES.EQ.0) CALL KPLOTC(STORE(J),W,Y1(IX),Y1(IY))
       IF (NXES.GT.0) CALL KPLOTC(STORE(J),W,Y1(IY),Y1(IX))
2400   READ (NTAPE) Y1
       IF (EOF(NTAPE)) 2800,2700

C      SKIP THIS PART IF PLOTTING FROM A FILE
2500   DO 2600 I=NDIM1,NDIM2
       IF (NXES.EQ.0) CALL KPLOTC(STORE(J),W,X(I),Y(I))
2600   IF (NXES.GT.0) CALL KPLOTC(STORE(J),W,Y(I),X(I))

C      RESUME HERE
2700   IF ((IST-10*(IST/10)).GT.0) CALL QPRINT(W,NO,NCOUNT,STORE)
       RETURN

C      ERROR MESSAGE
2800   WRITE (NO,2900)
2900   FORMAT(//32H INSUFFICIENT DATA ON INPUT FILE,/,
     1 1H ,28H PLOTTING ROUTINE TERMINATED)
       RETURN
       END
```

```
SUBROUTINE ADJUST(XH1,XL1)
IF (XH1.EQ.XL1) XL1=0.9*XL1-10.0
A=IFIX(100.0+ALOG10(XH1-XL1))-100.0
XH1T=XH1*10.0**(1.0-A)
XL1T=XL1*10.0**(1.0-A)
IF (XH1T .GE. 0.0) XH1T=IFIX(XH1T+0.9)
XH1T=IFIX(XH1T)
IF (XL1T .LE. 0.0) XL1T=IFIX(XL1T-0.9)
XL1T=IFIX(XL1T)
XH1=XH1T*10.0**(A-1.0)
XL1=XL1T*10.0**(A-1.0)
RETURN
END
```

```
      SUBROUTINE QINIT(IMAGE)
      COMMON /PLOT1/
     1 NV,NH,NCPW,LW,Q(4),NXES,NDIR,IST,NGLV,NGLH,BSYM,GSYM
      DIMENSION IMAGE(1)
      DATA IBLNK/10H          /
      N=LW*NV
      DO 100 I=1,N
100   IMAGE(I)=IBLNK
      DO 101 I=1,NH
      CALL QPLOT(IMAGE,I,1,BSYM)
101   CALL QPLOT(IMAGE,I,NV,BSYM)
      DO 102 I=1,NV
      CALL QPLOT(IMAGE,1,I,BSYM)
102   CALL QPLOT(IMAGE,NH,I,BSYM)
1800  IF (NGLV.EQ.0) GO TO 2000
      NGLV1=NGLV+1
      NH1=NH-1
      DO 1900 I=NGLV1,NH1,NGLV
      DO 1900 J=1,NV
1900  CALL QPLOT(IMAGE,I,J,GSYM)
2000  IF (NGLH.EQ.0) RETURN
      NGLH1=NGLH+1
      NV1=NV-1
      DO 2100 I=NGLH1,NV1,NGLH
      DO 2100 J=1,NH
2100  CALL QPLOT(IMAGE,J,I,GSYM)
      RETURN
      END
```

```
      SUBROUTINE KPLOTC(W,IMAGE,X,Y)

      DIMENSION W(1), IMAGE(1)
      COMMON /PLOT1/ NV, NH

      J=(X-W(2))*W(6)+1.5
      IF ((J.LE.0).OR.(J.GT.NH)) RETURN
      I=NV-IFIX((Y-W(4))*W(7)+0.5)
      IF ((I.LE.0).OR.(I.GT.NV)) RETURN
      CALL QPLOT (IMAGE,J,I,W(1))
      RETURN
      END
```

```
      SUBROUTINE QPLOT(IMAGE,J,I,SYM)

      COMMON /PLOT1/ NV, NH, NCPW, LW
      DIMENSION IMAGE(1)

      II=J/NCPW
      L=J-NCPW*II
      II=II+1
      IF (L) 101,101,102
101   L=NCPW
      II=II-1
102   IW=II+(I-1)*LW
      CALL PLACE(IMAGE(IW),L,SYM,1)
103   RETURN
      END
```

```
      SUBROUTINE PLACE(A,N,B,M)
C     THE MTH CHAR OF B REPLACES
C     THE NTH CHARACTER OF A
C     CHAR POSITIONS ARE 1 TO 10 FROM LEFT TO RIGHT

      COMMON/INOU/KIN,KOUT
      INTEGER A, B, BX, BY
      DATA MASK/77B/

C     CHECK FOR VALID ARGUMENTS
      IF (N.GT.10 .OR. M.GT.10) GO TO 900
      IF (N.LT.1 .OR. M.LT.1) GO TO 900

C     NULL ALL BUT THE MTH CHAR OF B, PUT IT IN BX
C     NULL THE NTH CHAR OF A
      NSHFT=60-6*N
      MSHFT=60-6*M
      MASKBY = SHIFT(MASK,NSHFT)
      MASKB = SHIFT(MASK,MSHFT)
      MASKA = COMPL(MASKBY)
      A = AND(A,MASKA)
      BX = AND(B,MASKB)

C     SHIFT THE MTH CHAR OF BX TO THE NTH POSITION
C     AND PUT IT IN BY
      MNSHFT=6*(M-N)
      BY = SHIFT(BX,MNSHFT)
      BY = AND(BY,MASKBY)

C     COMBINE A AND BY
      A = OR(A,BY)

      RETURN

C     N OR M OUT OF BOUNDS
900   WRITE (KOUT,1900)
1900  FORMAT (1H ,20HERROR IN SUBR. PLACE,/,
     1 24H N OR M IS OUT OF BOUNDS)
      CALL EXIT

      END
```

```
      SUBROUTINE QPRINT(IMAGE,NO,NCOUNT,STORE)

      DIMENSION IMAGE(1), STORE(1)
      COMMON /PLOT1/ NV, NH, NCPW, LW

      CALL PAGEFD(NO,1)
      DO 110 I=1,NCOUNT
      IB=7*(I-1)+1
110   WRITE (NO,102) STORE(IB+1),STORE(IB),STORE(IB),STORE(IB+2)
      NCANT=NV-NCOUNT
      IA=1
      DO 150 I=1,NV
      IB=I*LW
      IF (I.GT.NCOUNT) GO TO 120
      IBASE=(I-1)*7+1
      WRITE (NO,103) STORE(IBASE),STORE(IBASE+4),(IMAGE(J),J=IA,IB)
      GO TO 150
120   IF (I.GT.NCANT) GO TO 130
      WRITE (NO,105) (IMAGE(J),J=IA,IB)
      GO TO 150
130   IBASE=(I-1-NCANT)*7+1
      WRITE (NO,103) STORE(IBASE),STORE(IBASE+3),(IMAGE(J),J=IA,IB)
150   IA=IA+LW
102   FORMAT(1H ,11X,1PE10.3,1X,A1,57X,A1,1X,1PE10.3)
103   FORMAT(1H ,A1,1PE9.2,1X,12A10,A1)
105   FORMAT(1H ,11X,12A10,A1)
      RETURN
      END
```

```
C       LIBRARY FOR OPTIMAL CONTROL MODEL

        SUBROUTINE KFLTR(N,M,A,HT,Q,R,X,G,Z)
C          SOLVES X=A(X-XH'(HXH'+R)HX)A'+Q
        DIMENSION A(1), HT(1), Q(1), R(1), X(1), G(1), Z(1), TR(80)
        COMMON /MAIN1/ NDIM, NDIM1, F(1)
        COMMON /INOU/  KIN, KOUT

        NN=N*NDIM
        NT=1
        II=0
        DO 1 I=1,N
        IF (2**NT.GT.N) GO TO 2
1       NT=NT+1
2       DO 4 J=1,M
        DO 3 I=1,N
        K=II+I
3       G(K)=HT(K)/R(J)
4       II=II+NDIM
        CALL EQUATE(X,A,N,N)
        CALL MAT2(N,M,G,HT,Z)
        DO 15 L=1,NT
        CALL MAT3A(N,N,X,Z,F)
        DO 10 I=1,N
        DO 10 J=I,NN,NDIM
        Z(J)=Z(J)+F(J)
10      F(J)=X(J)
15      CALL MMUL(F,F,N,N,N,X)
        CALL FACTOR(N,Z,F,MR)
        IF (MR.EQ.-1) CALL EXIT
        CALL GMINV(N,N,F,Z,MR,0)
        WRITE (KOUT,16) N, N, MR
16      FORMAT (25HOOBSERVABILITY MATRIX IS ,I2,1HX,I2,9H OF RANK ,I2)
        CALL MMUL(X,Z,N,N,N,F)
        CALL MAT2(N,N,F,F,X)
        ENTRY KFLTR1
        TOL=1.E-5
        ADV=TOL*1.E-7
        NN=N*NDIM
        L=0
        TOL1=TOL/10.
        MAXIT=30
        DO 19 I=1,N
        TR(I)=-1.0
19      CONTINUE
        DO 40 IT=1,MAXIT
        CALL MMUL(X,HT,N,N,M,F)
        CALL MAT2A(N,M,HT,F,G)
        II=1
        DO 20 I=1,M
        G(II)=G(II)+R(I)
20      II=II+NDIM1
        CALL GMINV(M,M,G,Z,MR,0)
        CALL MMUL(F,Z,N,M,M,G)
        CALL MAT5(G,HT,N,M,N,Z)
        II=1
        DO 18 I=1,N
        DO 17 J=I,NN,NDIM
17      Z(J)=-Z(J)
        Z(II)=Z(II)+1.0
18      II=II+NDIM1
        IF (L.EQ.N) RETURN
        CALL MMUL(A,Z,N,N,N,X)
        CALL MMUL(A,G,N,N,M,F)
        II=0
```

```
      DO 22 J=1,M
      DO 21 I=1,N
      K=II+I
21    G(K)=F(K)*R(J)
22    II=II+NDIM
      CALL MAT2(N,M,G,F,Z)
      CALL MADD1(N,N,Z,Q,Z,1.0)
      CALL TRANS1(N,X,X)
      CALL DLINEQ(N,X,Z,X,TOL1)
      L=0
      C1=0.0
      II=1
      DO 25 I=1,N
      IF (ABS(X(II)-TR(I)).LT.(ADV+TOL*X(II))) L=L+1
      TR(I)=X(II)
      II=II+NDIM1
25    C1=C1+TR(I)
      IF (ABS(C1) .GT. 1.0E20) GO TO 50
      IF (L.NE.N) GO TO 40
      WRITE (KOUT,27) IT
27    FORMAT(17HORICCATI SOLN IN ,I2,11H ITERATIONS)
      CALL GMINV(N,N,Z,F,MR,0)
      IF (MR.NE.N) WRITE (KOUT,35) MR
35    FORMAT(26HORICCATI SOLN IS PSD--RANKI3)
40    CONTINUE
      WRITE (KOUT,45) MAXIT
45    FORMAT(26HORICCATI NON-CONVERGENT INI2,11H ITERATIONS)
      CALL EXIT
50    WRITE (KOUT,55) IT, NT
55    FORMAT(29HORICCATI BLOW UP AT ITERATIONI2,12H  INITIAL N=I3)
      CALL EXIT
      RETURN
      END
```

```
      SUBROUTINE MLINEQ(N,A,C,X,TOL1)
C     ANSWER RETURNED IN C AND X
      DIMENSION A(1),C(1),X(1)
      COMMON /MAIN1/ NDIM, NDIM1, F(1)
      COMMON /INOU/  KIN, KOUT

      DT=.5
      DT1=0.
      NN=N*NDIM
      DO 5 II=1,NN,NDIM1
5     DT1=DT1+ABS(A(II))
      DT1=DT1/N
      IF (DT1.GT.4.0) DT=DT*4.0/DT1
      II=1
      DO 20 I=1,N
      DO 15 JJ=I,NN,NDIM
15    X(JJ)=DT*A(JJ)
      X(II)=X(II)-.5
20    II=II+NDIM1
      CALL GMINV(N,N,X,F,MR,1)
      IF (MR.EQ.N) GO TO 21
      IT=0
      DO 18 I=1,NN,NDIM1
18    C(I)=1.E25
      GO TO 95
21    CALL MMUL(C,F,N,N,N,X)
C     INITIALIZATION OF X,F
      I=1
      DO 40 II=1,NN,NDIM
      J=II
      IF (I.EQ.1) GO TO 30
      DO 25 JJ=I,II,NDIM
      C(J)=C(JJ)
25    J=J+1
30    ID=J
      DO 35 JJ=II,NN,NDIM
      C(J)=DT*DOT(N,F(II),X(JJ))
35    J=J+1
      F(ID)=F(ID)+1.0
40    I=I+1
      GO TO 50
      ENTRY DLINEQ
      NN=N*NDIM
      CALL EQUATE(F,A,N,N)
50    TOL=TOL1
      ITT=31
      IF (TOL1 .GT. 0.99) ITT=IFIX(TOL)
      IF (TOL1 .GT. 0.99) TOL=0.0
      ADV=TOL*1.E-7
      DO 90 IT=1,ITT
      NEZ=0
      SIZE=0.0
      CALL MMUL(C,F,N,N,N,X)
      I=1
      II=1
      J=1
      GO TO 70
60    J=II
      DO 65 JJ=I,II,NDIM
      C(J)=C(JJ)
65    J=J+1
70    ID=J
      DT1=C(J)
      DO 75 JJ=II,NN,NDIM
      C(J)=C(J)+DOT(N,F(II),X(JJ))
75    J=J+1
```

```
      J=J-1
      DO 80 JJ=II,J
80    X(JJ)=F(JJ)
      IF (ABS(C(ID)-DT1).LT.(ADV+TOL*ABS(C(ID)))) NEZ=NEZ+1
      I=I+1
      II=II+NDIM
      SIZE=SIZE+DT1
      IF (I.LE.N) GO TO 60
      IF (NEZ.EQ.N) GO TO 150
      IF (ABS(SIZE) .GT. 1.0E18) GO TO 95
      CALL MMUL(X,X,N,N,N,F)
      IF (IT.EQ.ITT) GO TO 150
90    CONTINUE
95    WRITE (KOUT,100) IT
100   FORMAT (33HOLIN EQN ALGORITHM NON-CONVERGENT,I3,10HITERATIONS)
150   CALL EQUATE(X,C,N,N)
      RETURN
      END
```

```
      SUBROUTINE DINTEG(N,A,C,S,NT)
C      S=SUM I=0 TO NT-1 OF A^I*C*AT^I

      DIMENSION A(1), C(1), S(1)

      COMMON
     1 /MAIN1/ NDIM, NDIM1, COM1(1)
     2 /MAIN2/ COM2(1)

      K=NT
      NMX=(N-1)*NDIM+N
      DO 10 I=1,N
      DO 10 J=I,NMX,NDIM
      S(J)=0.0
      COM2(J)=C(J)
10    CONTINUE

      IFLAG=0
100   IP=MOD(K,2)
      K=K/2
      IF (IP.EQ.0) GO TO 110
      IF (IFLAG.EQ.0) CALL EQUATE(COM1,A,N,N)
      IF (IFLAG.EQ.1) CALL MMUL(A,C,N,N,N,COM1)
      IFLAG=1

      DO 14 I=1,N
      DO 14 J=I,NMX,NDIM
      C(J)=COM1(J)
      S(J)=S(J)+COM2(J)
14    CONTINUE
      IF (K.EQ.0) GO TO 200

      CALL MMUL(A,COM2,N,N,N,COM1)
      CALL MAT2(N,N,COM1,A,COM2)
110   CALL MMUL(A,COM2,N,N,N,COM1)
      CALL MAT6S(N,N,COM1,A,COM2)
      CALL MMUL(A,A,N,N,N,COM1)
      CALL EQUATE(A,COM1,N,N)
      GO TO 100

200   RETURN
      END
```

```
      SUBROUTINE DSCRT(N,A,DEL,EA,EAINT,NT)
      DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C         SETS EA=EXP(A*DEL),EAINT=INTEGRAL EA 0 TO DEL
      COMMON/MAIN1/NDIM,NDIM1
      NN=N*NDIM
      NTM1=NT-1
      COEF(NT)=1.
      DO 10 I=1,NTM1
      II=NT-I
10    COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C                   NT MUST BE AT LEAST 3
      II=1
      DO 30 I=1,N
      DO 20 J=I,NN,NDIM
20    EAINT(J)=A(J)*COEF(1)
      EAINT(II)=EAINT(II)+COEF(2)
30    II=II+NDIM1
      DO 60 L=3,NT
      T1=COEF(L)
      CALL MMUL(A,EAINT,N,N,N,EA)
      IF(L.EQ.NT)GO TO 70
      II=1
      DO 60 I=1,N
      DO 50 J=I,NN,NDIM
50    EAINT(J)=EA(J)
      EAINT(II)=EAINT(II)+T1
60    II=II+NDIM1
70    DO 80 II=1,NN,NDIM1
      EA(II)=EA(II)+T1
80    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FACTOR(N,A,S,MR)
C            A=ST*S
      DIMENSION A(1), S(1), IP(80)
      COMMON /MAIN1/ NDIM, NDIM1
      COMMON /INOU/  KIN, KOUT
      MR=0
      NN=N*NDIM
      TOL=1.0E-06
      CALL MSCALE(S,A,N,N,1.0)
      I=1
      DO 50 ID=1,NN,NDIM1
      NEND=ID-I
      R=0.0
      IP(I)=I
      K=I
      DO 20 KK=ID,NN,NDIM1
      T=S(KK)-DOT2(NEND,S(K),S(K))
      IF (ABS(T) .LE. (TOL*(ABS(S(KK))+TOL))) GO TO 20
      IF (ABS(T) .LE. ABS(R)) GO TO 20
      IP(I)=K
      R=T
20    K=K+1
      IF (I .EQ. IP(I)) GO TO 25
      K=IP(I)
      CALL SWAP(S(I),S(K),N,NDIM)
      K=ID+NDIM*(K-I)
      NP=N-I+1
      CALL SWAP(S(ID),S(K),NP,1)
25    NP=ID
      IF (R) 30, 35, 40
30    MR=-1
      WRITE (KOUT,1000)
1000  FORMAT (1H ,33HTRIED TO FACTOR INDEFINITE MATRIX)
      RETURN
35    NP=NEND+N
40    K=NEND+1
      DO 42 KK=K,NP
42    S(KK)=0.0
      IF (R .EQ. 0.0) GO TO 50
      T=SQRT(R)
      S(ID)=T
      MR=MR+1
      IF (I .EQ. N) GO TO 55
      ID1=ID+1
      K=I+1
      NP=NEND+N
      DO 43 KK=ID1,NP
      S(KK)=(S(KK)-DOT2(NEND,S(I),S(K)))/T
43    K=K+1
50    I=I+1
C     UNSWAP ROWS
55    DO 60 I=1,N
      K=N-I+1
      KK=IP(K)
      IF (KK .EQ. K) GO TO 60
      CALL SWAP(S(K),S(KK),MR,NDIM)
60    CONTINUE
      CALL TRANS1(N,S,S)
      RETURN
      END
```

```
      SUBROUTINE GMINV(NR,NC,A,U,MR,MT)
      DIMENSION A(1),U(1),S(30)
      COMMON/MAIN1/ NDIM,NDIM1
      COMMON/INOU/  KIN, KOUT
      TOL=1.E-12
      MR=NC
      NRM1=NR-1
      TOL1=1.E-20
      JJ=1
      DO 100 J=1,NC
      FAC=DOT(NR,A(JJ),A(JJ))
      JM1=J-1
      JRM=JJ+NRM1
      JCM=JJ+JM1
      DO 20 I=JJ,JCM
20    U(I)=0.
      U(JCM)=1.0
      IF(J.EQ.1) GO TO 54
      KK=1
      DO 30 K=1,JM1
      IF(S(K).EQ.1.0) GO TO 30
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
30    KK=KK+NDIM
      DO 50 L=1,2
      KK=1
      DO 50 K=1,JM1
      IF(S(K).EQ.0.) GO TO 50
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(NR,TEMP,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
50    KK=KK+NDIM
      TOL1=TOL*FAC
      FAC=DOT(NR,A(JJ),A(JJ))
54    IF(FAC.GT.TOL1) GO TO 70
      DO 55 I=JJ,JRM
55    A(I)=0.
      S(J)=0.
      KK=1
      DO 65 K=1,JM1
      IF(S(K).EQ.0.) GO TO 65
      TEMP=-DOT(K,U(KK),U(JJ))
      CALL VADD(NR,TEMP,A(JJ),A(KK))
65    KK=KK+NDIM
      FAC=DOT(J,U(JJ),U(JJ))
      MR=MR-1
      GO TO 75
70    S(J)=1.0
      KK=1
      DO 72 K=1,JM1
      IF(S(K).EQ.1.) GO TO 72
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
72    KK=KK+NDIM
75    FAC=1./SQRT(FAC)
      DO 80 I=JJ,JRM
80    A(I)=A(I)*FAC
      DO 85 I=JJ,JCM
85    U(I)=U(I)*FAC
100   JJ=JJ+NDIM
      IF(MR.EQ.NR.OR.MR.EQ.NC) GO TO 120
      IF(MT.NE.0) WRITE (KOUT,110) NR,NC,MR
110   FORMAT(I3,1HX,I2,8H M  RANK,I2)

120   NEND=NC*NDIM
      JJ=1
```

```
      DO 135 J=1,NC
      DO 125 I=1,NR
      II=I-J
      S(I)=0.
      DO 125 KK=JJ,NEND,NDIM
125   S(I)=S(I)+A(II+KK)*U(KK)
      II=J
      DO 130 I=1,NR
      U(II)=S(I)
130   II=II+NDIM
135   JJ=JJ+NDIM1
      RETURN
      END
```

```
      SUBROUTINE MAT2(N1,N2,X,Y,Z)
C     Z=XY' X,Y=N1*N2,Z=Z'
C     Z AND Y CAN BE EQUIVALENT
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN2=N2*NDIM
      II=1
      DO 10 I=1,N1
      IJ=II
      DO 5 J=I,N1
      Z(IJ)=DOT2(NN2,X(I),Y(J))
5     IJ=IJ+NDIM
      J=II
      IJ=J
3     IJ=IJ-NDIM
      IF(IJ.LT.I) GO TO 10
      J=J-1
      Z(IJ)=Z(J)
      GO TO 3
10    II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE MAT2A(N1,N2,X,Y,Z)
C       Z=XT*Y  X,Y ARE N1XN2,  Z=ZT
C      Z AND Y CAN BE EQUIVALENT
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      NN2=N2*NDIM
      I=0
      DO 10 II=1,NN2,NDIM
       J=II+I
       IJ=J-1
       DO 5 JJ=II,NN2,NDIM
       Z(J)=DOT(N1,X(II),Y(JJ))
5      J=J+1
       I=I+1
       JJ=I
       DO 10 J=II,IJ
       Z(J)=Z(JJ)
10     JJ=JJ+NDIM
       RETURN
       END
```

```
      SUBROUTINE MAT3(N1,N2,X,Y,Z)
C      Z=XYXT   Y=YT IS N2XN2     X IS N1XN2
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/ NDIM
      CALL MMUL(X,Y,N1,N2,N2,Z)
      NN2=N2*NDIM
      DO 5 I=1,N1
      IM1=I-1
      II=IM1*NDIM
      JJ=I+II
      DO 3 J=I,N1
      Z(JJ)=DOT2(NN2,Z(J),X(I))
3     JJ=JJ+NDIM
      JJ=I
      K=II+1
      KK=II+IM1
      DO 5 J=K,KK
      Z(JJ)=Z(J)
      JJ=JJ+NDIM
5     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MAT3A(N1,N2,X,Y,Z)
C     Z=X'YX  Y=Y' IS N2XN2  X' IS N1XN2
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN1=N1*NDIM
      CALL MMUL(Y,X,N2,N2,N1,Z)
      I=0
      DO 10 II=1,NN1,NDIM
      J=II+I
      IJ=J-1
      DO 5 JJ=II,NN1,NDIM
      Z(J)=DOT(N2,X(II),Z(JJ))
5     J=J+1
      I=I+1
      JJ=I
      DO 10 J=II,IJ
      Z(J)=Z(JJ)
10    JJ=JJ+NDIM
      RETURN
      END
```

```
      SUBROUTINE MAT5(X,Y,N1,N2,N3,Z)
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM,NDIM1
C        Z=X*YT    X IS N1*N2, Y IS N3*N2
      JJ=1
      DO 3 I=1,N3
      C1=Y(I)
      CALL VSCALE(Z(JJ),X,N1,C1)
3     JJ=JJ+NDIM
      K=NDIM1
      GO TO 4
      ENTRY MAT5S
      K=1
4     NN2=N2*NDIM+1
      NN3=N3*NDIM
7     IF(K.EQ.NN2) RETURN
      I=K
      DO 6 JJ=1,NN3,NDIM
      C1=Y(I)
      IF(C1.NE.0.) CALL VADD(N1,C1,Z(JJ),X(K))
6     I=I+1
      K=K+NDIM
      GO TO 7
      END
```

```
      SUBROUTINE MAT5A(X,Y,N1,N2,N3,Z)
C         Z=XT*Y    X=N2*N1, Y=N2*N3
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      N1M1=N1-1
      NN3=N3*NDIM
      DO 1 I=1,NN3,NDIM
      II=I+N1M1
      DO 1 J=I,II
1     Z(J)=0.0
      ENTRY MAT5AS
      NN3=N3*NDIM
      DO 10 K=1,N2
      KK=K
      DO 8 I=1,N1
      C1=X(KK)
      IF(C1.NE.0.0) CALL VADD1(NN3,C1,Z(I),Y(K))
8     KK=KK+NDIM
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MAT6(N1,N2,X,Y,Z)
C     Z=X*Y',  WHERE X=N1*N2, Y=N1*N2, Z=Z'=N1*N1
      DIMENSION X(1), Y(1), Z(1)
      COMMON /MAIN1/ NDIM, NDIM1
      NN1=N1*NDIM
      DO 1 I=1,N1
      DO 1 J=I,NN1,NDIM
      Z(J)=0.0
1     CONTINUE
      ENTRY MAT6S
C     Z=Z+X*Y'
      NN2=N2*NDIM
      NN1=N1*NDIM
      DO 6 K=1,NN2,NDIM
      KK=K-1
      J=1
      DO 6 I=1,N1
      C1=Y(I+KK)
      IF (C1.NE.0.0) CALL VADD(I,C1,Z(J),X(K))
      J=J+NDIM
6     CONTINUE
      IF (N1.EQ.1) RETURN
      NN2=NDIM1+1
      DO 10 K=NN2,NN1,NDIM1
      I=K
      J=K
8     I=I-1
      J=J-NDIM
      Z(J)=Z(I)
      IF (J.GT.NDIM) GO TO 8
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MMUL(X,Y,N1,N2,N3,Z)
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      N1M1=N1-1
      NN3=N3*NDIM
      DO 1 I=1,NN3,NDIM
      II=I+N1M1
      DO 1 J=I,II
1     Z(J)=0.0
      ENTRY MMULS
      NN3=N3*NDIM
      KK=0
      DO 10 K=1,N2
      DO 8 I=1,N1
      C1=X(I+KK)
      IF(C1.NE.0.0) CALL VADD1(NN3,C1,Z(I),Y(K))
8     CONTINUE
10    KK=KK+NDIM
      RETURN
      END
```

```
      SUBROUTINE MADD1(NR,NC,X,Y,Z,C1)
C     Z=X+C1*Y
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      NN=NC*NDIM
      IF(C1.EQ.1.0)GO TO 1
      IF(C1.EQ.-1.)GO TO 2
      DO 5 I=1,NR
      DO 5 J=I,NN,NDIM
5     Z(J)=X(J)+C1*Y(J)
      RETURN
1     DO 10 I=1,NR
      DO 10 J=I,NN,NDIM
10    Z(J) = X(J)+Y(J)
      RETURN
2     DO 15 I=1,NR
      DO 15 J=I,NN,NDIM
15    Z(J)=X(J)-Y(J)
      RETURN
      END
```

```
      SUBROUTINE DIAG2(N,A,B,C1,C2)
C     A = C1*B + C2*I
C     A,B ARE N*N MATRICES; I IS N*N IDENTITY MATRIX
      DIMENSION A(1), B(1)
      COMMON /MAIN1/ NDIM, NDIM1
      NN=N*NDIM
      NM1=N-1
      II=1
      IF (C1 .EQ. 1.0) GO TO 10
      DO 5 J=1,NN,NDIM
      K=J+NM1
      DO 4 I=J,K
4     A(I)=C1*B(I)
      A(II)=A(II)+C2
5     II=II+NDIM1
      RETURN
10    DO 7 J=1,NN,NDIM
      K=J+NM1
      DO 6 I=J,K
6     A(I)=B(I)
      A(II)=A(II)+C2
7     II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE IDENT(N,A,C1)
      DIMENSION A(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN=N*NDIM
      II=1
      DO 1 I=1,N
      DO 2 J=I,NN,NDIM
2     A(J)=0.0
      A(II)=C1
1     II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE EQUATE(A,B,NR,NC)
C     A=B
C     MATRIX EQUATE
      DIMENSION A(1), B(1)
      CALL MSCALE(A,B,NR,NC,1.0)
      RETURN
      END
```

AD-A045 036 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS F/G 9/2
TIVAR: A COMPUTER PROGRAM FOR PREDICTING ENSEMBLE STATISTICS IN--ETC(U)
JUL 77 D L KLEINMAN, S BARON, J E BERLINER DAAH01-76-C-0194
UNCLASSIFIED BBN-3464 DRDMI-TD-CR-77-4 NL

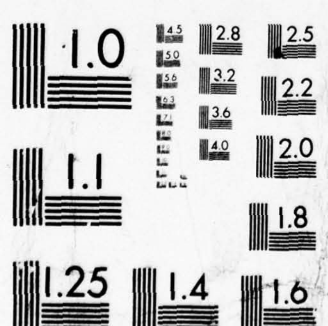
2 OF 2
AD
A045036

END
DATE
FILMED
11-77
DDC

2 OF 2

D

045036



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      SUBROUTINE MSCALE(A,B,NR,NC,C1)
C     A=C1*B
C     A AND B MAY BE EQUIVALENT
      DIMENSION A(1), B(1)
      COMMON /MAIN1/ NDIM
      NN=NC*NDIM
      IF (C1 .EQ. 1.0) GO TO 10
      IF (C1 .EQ. 0.0) GO TO 20
      IF (C1 .EQ. -1.) GO TO 30
      DO 5 I=1,NR
      DO 5 J=I,NN,NDIM
5     A(J)=C1*B(J)
      RETURN
10    DO 15 I=1,NR
      DO 15 J=I,NN,NDIM
15    A(J)=B(J)
      RETURN
20    DO 25 I=1,NR
      DO 25 J=I,NN,NDIM
25    A(J)=0.0
      RETURN
30    DO 35 I=1,NR
      DO 35 J=I,NN,NDIM
35    A(J)=-B(J)
      RETURN
      END
```

```
      SUBROUTINE TRANS1(N,A,AT)
C     SETS AT = ATRANSPOSE.
C     A AND AT CAN BE EQUIVALENT; BOTH ARE SQUARE.
      DIMENSION A(1),AT(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN=N*NDIM
      DO 1 I=1,NN,NDIM1
      IJ=I
      DO 1 JI=I,NN,NDIM
      TEMP=A(IJ)
      AT(IJ)=A(JI)
      AT(JI)=TEMP
      IJ=IJ+1
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRANSP(N1,N2,X,XPOSE)
C     COMPUTE MATRIX TRANSPOSE
C     XPOSE=X'
C     X AND XPOSE CANNOT BE EQUIVALENT
      DIMENSION X(1),XPOSE(1)
      COMMON /MAIN1/NDIM
      ENTRY TRANS2
      K=1-NDIM
      DO 10 I=1,N1
      K=K+NDIM
      IJ=I
      JI=K
      DO 10 J=1,N2
      XPOSE(JI)=X(IJ)
      IJ=IJ+NDIM
      JI=JI+1
10    CONTINUE
      RETURN
      END
```

```
      FUNCTION DOT(NR,A,B)
      DOUBLE PRECISION DDT1, DBLE
      DIMENSION A(1),B(1)
      DDT1=0.0D0
      IF (NR .LE. 0) GO TO 2
      DO 1 I=1,NR
1     DDT1=DDT1+DBLE(A(I)*B(I))
2     DOT=DDT1
      RETURN
      END
```

```
      FUNCTION DOT2(NN,A,B)
      DOUBLE PRECISION DDT2, DBLE
      DIMENSION A(1),B(1)
      COMMON /MAIN1/ NDIM
      DDT2=0.0D0
      IF (NN .LE. 0) GO TO 2
      DO 1 I=1,NN,NDIM
1     DDT2=DDT2+DBLE(A(I)*B(I))
2     DOT2=DDT2
      RETURN
      END
```

```
      FUNCTION DOT3(N,A,B)
      DOUBLE PRECISION DDT3, DBLE
      DIMENSION A(1),B(1)
      COMMON /MAIN1/ NDIM
      DDT3=0.0D0
      IF (N .LE. 0) GO TO 2
      II=1
      DO 1 I=1,N
      DDT3=DDT3+DBLE(A(II)*B(I))
1     II=II+NDIM
2     DOT3=DDT3
      RETURN
      END
```

```
      SUBROUTINE VADD(N,C1,A,B)
      DIMENSION A(1),B(1)
      DO 1 I=1,N
1     A(I)=A(I)+C1*B(I)
      RETURN
      END
```

```
      SUBROUTINE VADD1(NN,C1,A,B)
      DIMENSION A(1),B(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,NN,NDIM
1     A(I)=A(I)+C1*B(I)
      RETURN
      END
```

```
      SUBROUTINE VSCALE(X,Y,N,C1)
      DIMENSION X(1),Y(1)
      L=0
      IF(C1.EQ.1.0) GO TO 5
      IF(C1.EQ.0.0) GO TO 8
      IF(C1.EQ.-1.) GO TO 13
1     L=L+1
      X(L)=C1*Y(L)
      IF(L.LT.N) GO TO 1
      RETURN
5     L=L+1
      X(L)=Y(L)
      IF(L.LT.N) GO TO 5
      RETURN
8     L=L+1
      X(L)=0.0
      IF(L.LT.N) GO TO 8
      RETURN
13    L=L+1
      X(L)=-Y(L)
      IF(L.LT.N) GO TO 13
      RETURN
      END
```

```
      SUBROUTINE SWAP(A,B,N,INC)
C     SWAPS A ROW, COLUMN, OR DIAGONAL OF TWO MATRICES
C     INC=1         ROW
C     INC=NDIM      COLUMN
C     INC=NDIM+1    DIAGONAL
      DIMENSION A(1), B(1)
      NN=N*INC
      I=1
1     IF (I .GT. NN) RETURN
      TEMP=A(I)
      A(I)=B(I)
      B(I)=TEMP
      I=I+INC
      GO TO 1
      END
```

```
      SUBROUTINE VMAT1(A,X,N1,N2,Y)
C     Y=AX
      DIMENSION A(1),X(1),Y(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,N1
      Y(I)=0.0
      II=I
      DO 1 J=1,N2
      Y(I)=Y(I)+A(II)*X(J)
1     II=II+NDIM
      RETURN
      END
```

```
      SUBROUTINE VMAT2(Z,A,X,N1,N2,Y)
C     Y=Z+AX
      DIMENSION A(1),X(1),Z(1),Y(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,N1
      Y(I)=Z(I)
      II=I
      DO 1 J=1,N2
      Y(I)=Y(I)+A(II)*X(J)
1     II=II+NDIM
      RETURN
      END
```

```
      FUNCTION XGAIN(TH,XM,XS)
      DIMENSION A(5)
      DATA A/.2258368,-.2521287,1.259695,-1.287822,.9406461/
      IF (TH.GT.0.) GO TO 2
      XGAIN=1.0
      RETURN
2     Y=XM
      NS=2
      IF(XS.LT.1.0E-10)XS=1.0E-10
      IF(Y.EQ.0.) NS=1
      ANS=0.
      RMS=XS**2+XM**2
      DO 1 I=1,NS
      Z=.707*(TH+Y)/XS
      TEMP=EXP(-Z**2)
      X=1./(1.+.327591*ABS(Z))
      P=X*((((A(5)*X+A(4))*X+A(3))*X+A(2))*X+A(1))*1.128379
      ERF=1.-P*TEMP
      IF (Z.LT.0.) ERF=-ERF
      ANS=ANS+(RMS+TH*Y)*(1.-ERF)-XS*Y*TEMP*.7975
1     Y=-Y
      XGAIN=ANS/RMS/FLOAT(NS)
      IF(XGAIN.LT.1.E-6) XGAIN=1.E-6
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(M,N,X,LA)
C     MATRIX OUTPUT ROUTINE (WITH HEADER)
      DIMENSION X(1)
      COMMON /MAIN1/ NDIM
      COMMON /INOU/  KIN, KOUT, KPTR
      DATA KB/1H /
      ENTRY MATTYP
      KFIL=KOUT
      GO TO 100
      ENTRY MATPRT
      CONTINUE
      ENTRY OUTMAT
      KFIL=KPTR
100   IF (LA .NE. KB) WRITE (KFIL,1000) LA
1000  FORMAT(/,1H ,A10,8H MATRIX )
C     WRITE THE MATRIX
      IO=3
      CALL MATIO(X,M,N,IO)
      RETURN
      END
```

```
      SUBROUTINE VECOUT(X,N,LET)
C     VECTOR OUTPUT ROUTINE (WITH HEADER)

      DIMENSION X(1)
      COMMON /MAIN1/ NDIM
      COMMON /INOU/  KIN, KOUT, KPTR
      DATA KB /1H /

      ENTRY VECTYP
      KFIL=KOUT
      GO TO 100
      ENTRY VECPRT
      KFIL=KPTR
100   IF (LET .NE. KB) WRITE (KFIL,1000) LET
1000  FORMAT (/,1H ,A10,8H VECTOR )

C     WRITE THE VECTOR
      IO=3
      CALL VECTIO(X,N,IO)

      RETURN
      END
```

```
      SUBROUTINE MATIO(X,NR,NC,IO)
C     BATCH ORIENTED MATRIX I/O
C     IO=1   INPUT ONLY
C     IO=2   INPUT AND OUTPUT
C     IO=3   OUTPUT ONLY
C     IO=4   PUNCH

      DIMENSION X(1)
      COMMON /MAIN1/ NDIM
      COMMON /INOU/  KIN, KOUT, KPTR, KPUNCH

      JEND=NC*NDIM
      GO TO (5,5,20,40) IO

C*******INPUT
5     DO 10 I=1,NR
      READ (KIN,1000) (X(IJ), IJ=I,JEND,NDIM)
10    CONTINUE
      IF (IO .EQ. 1) RETURN

C*******OUTPUT
20    DO 30 I=1,NR
      WRITE (KOUT,2000) (X(IJ), IJ=I,JEND,NDIM)
30    CONTINUE
      RETURN

C*******PUNCH
40    DO 50 I=1,NR
      WRITE (KPUNCH,3000) (X(IJ),IJ=I,JEND,NDIM)
50    CONTINUE
      RETURN

1000  FORMAT (8E10.0)
2000  FORMAT (1H ,1P10E13.3)
3000  FORMAT (1P8E10.3)
      END
```

```
      SUBROUTINE VECTIO(X,N,IO)
C     BATCH ORIENTED VECTOR I/O
C     IO=1   INPUT ONLY
C     IO=2   INPUT AND OUTPUT
C     IO=3   OUTPUT ONLY
C     IO=4   PUNCH

      DIMENSION X(1)
      COMMON /INOU/  KIN, KOUT, KPTR, KPUNCH

      GO TO (10,10,20,40) IO

C*******INPUT
10    READ (KIN,1000) (X(I), I=1,N)
      IF (IO .EQ. 1) RETURN

C*******OUTPUT
20    WRITE (KOUT,2000) (X(I), I=1,N)
      RETURN

C*******PUNCH
40    WRITE (KPUNCH,3000) (X(I), I=1,N)
      RETURN

1000  FORMAT (8E10.0)
2000  FORMAT (1H ,1P10E13.3)
3000  FORMAT (1P8E10.3)
      END
```

```
      SUBROUTINE KVECIO(IX,N,IO)
C     BATCH ORIENTED INTEGER VECTOR I/O
C     IO=1  INPUT ONLY
C     IO=2  INPUT AND OUTPUT
C     IO=3  OUTPUT ONLY
C     IO=4  PUNCH

      DIMENSION IX(1)
      COMMON /INOU/  KIN, KOUT, KPTR, KPUNCH

      GO TO (10,10,20,40) IO

C*******INPUT
10    READ (KIN,1000) (IX(I),I=1,N)
      IF (IO .EQ. 1) RETURN

C*******OUTPUT
20    WRITE (KOUT,2000) (IX(I),I=1,N)
      RETURN

C*******PUNCH
40    WRITE (KPUNCH,1000) (IX(I), I=1,N)
      RETURN

1000  FORMAT (8I10)
2000  FORMAT (1H ,10I13)

      END
```

```
      SUBROUTINE PAGEFD(KFIL,KOUNT)
C     WRITES KOUNT FORMFEEDS (1 IN COL 1) ON FILE KFIL

      ENTRY FORMFD
100   IF (KOUNT.LE.0) RETURN

      DO 200 I=1,KOUNT
      WRITE (KFIL,1000)
1000  FORMAT (1H1)
200   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE LINEFD(KFIL,KOUNT)
C     WRITES KOUNT LINEFEEDS (SPACE IN COL 1) ON FILE KFIL

100   IF (KOUNT.LE.0) RETURN

      DO 200 I=1,KOUNT
      WRITE (KFIL,1000)
1000  FORMAT (1H )
200   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE DAYTIM(KFIL)
C     WRITES THE DATE AND THE TIME ON FILE KFIL

10    CALL TIME(LTIME)
      CALL DATE(LDATE)
      WRITE (KFIL,1000) LDATE, LTIME
1000  FORMAT(1H ,A10,2X,A10)
      RETURN

      END
```

APPENDIX - GNPROP

A1. PROBLEM DESCRIPTION

GNPROP is an interface program for the TIVAR package which provides a means for generating time-varying gains L_d by backwards integration of the Riccati equation. Although GNPROP runs in reverse-time, those gains are punched out on cards by GNPROP in forward time, and may be inserted directly into the TIVAR input deck. In this appendix the procedure for running the GNPROP is described.

A2. TIME VARIATIONS

In executing GNPROP, any of the input parameters can be changed by the user at any time t. As in TIVAR, there are two methods for changing parameters. The first is via card inputs; the second is via a user written subroutine called INTOLD. (This subroutine corresponds to the subroutine INTNEW in TIVAR, but because GNPROP operates in reverse-time, it is named INTOLD.)

A2.1 System Codes

In GNPROP, a maximum of 16 system elements may be changed at any given time step. Each element, or parameter, is identified by a unique alphanumeric code and/or an index number. Table A1 defines the system codes used. As in TIVAR, when a given parameter I is changed at time t, GNPROP sets a flag IFLAG(I) equal to 1 for one time step. The implication of the parameter change is then addressed via the internal logic in GNPROP. If there have been no parameter changes, the internal flags remain at their nominal zero value.

A2.2 Changing System Parameters

Parameters may be changed at time t via external or card inputs. The alphanumeric code(s) is used to identify the parameter(s) being changed at a selected time. The input required for each code is given in section A3.

Parameters can be changed periodically via an internal subroutine INTOLD. The user must supply his own code -- in FORTRAN IV -- to use this option. For any specified code index number, I, the subroutine INTOLD(KEY) is called once every NDT(I) time step with KEY=I. The 16 values of NDT are inputs to GNPROP. When INTOLD is called, IFLAG(KEY) is set to 1. The user must supply the manner in which the parameters are to be changed. If no FORTRAN code is supplied to update the parameters, no changes are made. Thus, GNPROP assumes that the "new" parameters are identical to the previously existing ones.

Table A1: PARAMETER CODES IN GNPROP

CODE	INDEX	DESCRIPTION
A	1	System A matrix
B	2	System B matrix
C	3	Output C matrix
D	4	Output D matrix
E	5	Noise matrix, E
QX	6	State weightings vector
QY	7	Output weightings vector
QU	8	Control weightings vector
QR	9	Control rate weightings vector
RICI	10	First guess at a riccati solution
QXI	11	State weightings vector for a discrete time
QYI	12	Output wieghtings vector for a discrete time
PRINT	13	Trigger to print the current gains
GNOUT	14	Trigger to punch the current gains
DUMMY	15-16	Dummy codes, inactive at present

A3. INPUT DECK SETUP

The input deck structure for GNPROP is discussed, along with the user-written routine INTOLD.

A3.1 Control Cards

There are five major control cards required by GNPROP.

Card 1 - Title Information

Column 1: blank

Columns 2-80: alphanumeric title information

Card 2 - Gain Output Mode

Card 3 - Dimension Information, 5I5 Format

Field 1: NX = number of system states

Field 2: NX1= number of noise shaping states

Field 3: NU = number of control inputs $\leq$ 4

Field 4: NW = number of random noise sources

Field 5: NY = number of displayed outputs

The size restrictions are:

$$NX + NU \leq NDIM$$
$$NY \leq NDIM$$

where NDIM is the array size in the DIMENSION statemments (15).

Card 4 - Time Information, 3E10.0 Format

Field 1: DEL = discrete time step interval (sec)

Field 2: TO = Initial time (sec)

Field 3: TEND = terminal time (sec)

TO and TEND are specified as if the program were running in forward time. They correspond to the same quantites in TIVAR.

Card 5 - NDT(I) for internal time breaks, 16I5 Format

The 16 fields are associated with the 16 parameter codes in GNPROP on a one-to-one basis. The I-th field is associated with code I. NDT(I) is the

frequency (number of time steps) at which subroutine INTOLD is called internally with KEY=I, starting at time TEND. Calling INTOLD with KEY=I sets flag IFLAG(I)=1 for one time step. The actual parameter value must be changed by user-written code. If no code is supplied, the associated parameters retain their previous values.

A3.2 External Parameter Cards

The remaining input cards are used to change system parameters via external read-in at specified times. The deck set-up follows the same standard form as for TIVAR.

Time Card - Cols. 1-4 Alphanumeric TIME

Cols. 11-20 Time of external break, E10.0 Format

Code Card - Cols. 1-5 One of the alphanumeric codes given in Table A1.

Parameter Cards - The new parameter values required by the associated code.

The sequence of Code Card followed by new parameter values is repeated for all items that the user wishes to change at the given time. To change parameters at another time, input a new time card with the time of the desired change, followed by a code card, parameters cards, code card, etc. When using external (card) updates, the following rules must be observed:

1. Time breaks must occur in decreasing order, since the program is running in reverse-time.

2. The code cards can occur in any order.

3. The parameter cards must immediately follow the associated code card.

4. Parameter cards must, be input, as the program expects to read in new values.

5. The last card in the deck is an end card, containing the alphanumeric END in cols. 1-3.

Table A2 gives the required input for each of the active codes, as well as the initial parameter values that are set internally by the program, prior to t=TEND.

Data is entered on the cards in 8E10.0 Format, i.e., in floating point fields of 10 columns with a *maximum* of 8 fields per card. The numbers may be either in fixed-point (decimal) format or in scientific (exponential) format with the exponent right-justified in the field. Matrices are entered one row at a time. If a row contains less than 8 entries, the remaining fields on the card are left blank. If a row contains more than 8 entries, continue on a second card for that row. A new row always begins on a new card. Vectors are entered in similar 8E10.0 format: the first entry in the first field, the second entry in the second field, etc.

TABLE A2: CARD DATA INPUTS AND INITIALIZATION

CODE	INDEX	INPUT DATA	INITIAL VALUE
A	1	A_{IJ} ; I=1,...,NX, J=1,...,NX	A=0
B	2	B_{IJ} ; I=1,...,NX, j=1,...,NU	B=0
C	3	C_{ij} ; i=1,...,NY, j=1,...,NX	C=0
D	4	D_{ij} ; i=1,...,NY, j=1...,NU	D=0
E	5	E_{ij} ; i=1,...,NX, j=1,...,NW	E=0
QX	6	Q_{xi} ; i=1,...,NX	Q_x=0
QY	7	Q_{yi} ; i=1,...,NY	Q_y=0
QU	8	Q_{ui} ; i=1,...,NU	Q_u=0
QR	9	Q_{ri} ; i=1,...,NU	Q_r=0
RICI	10	$PINC_{ij}$; i=1,...,NX+NU, j=1,...,NX+NU	PINC=0
QXI	11	QI_{xi} ; i=1,...,NX	QI_x=0
QYI	12	QI_{yi} ; i=1,...,NY	QI_y=0
PRINT	13	---	---
GNOUT	14	---	---

A4. GNPROP SUBROUTINES

The various subroutines that constitute the GNPROP package are discussed briefly, along with the named common blocks.

A4.1 Subroutine Descriptions

1. Subroutine GNPROP. Provides the major logic control, dimensioning declarations and initialization for the entire package. Virtually no computation is done in GNPROP. GNPROP reads the 5 control cards.
2. Subroutine DNDATE. Reads input code cards and the subsequent data input cards. Calls INTOLD periodically as per NSTEP. Analagous to subroutine UPDATE in the TIVAR program.
3. Subroutine INTOLD. User written routine for internal breaks.
4. Subroutine GPFBGN. Converts L_d to L_c and outputs equivalent T_N and L_{opt}.
5. Subroutine GNRITE. Outputs the gains as specified by the second control card.

A4.2 Common Block Usage

Named common blocks are used to store most data arrays and to pass information among the various subroutines. The dimension declarations are given in the primary subprogram GNPROP.

1. /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)

   NCODES = Number of possible system codes in Table A1 (16)
   ICODES = Alphanumeric code identifiers

   IFLAG = 0 or 1 flags to indicate parameter changes
   NSTEP = frequencies for internal breaks, control card 5

2. /INOU/ KIN, KOUT, KPTR, KPUNCH, KDISK

   Logical unit numbers for input/output devices

3. /MAIN1/ NDIM, NDIM1, COM1 /MAIN2/ COM2

   Common blocks required for library subroutines. NDIM = dimension of program square arrays.

4. /COMP1/, /COMP3/

   Common blocks used for internal computations and internal storage of temporary matrices.

5. /INPTS/ BD, ADT

   Discretized system variables
   ADT = discrete system augmented A matrix (transpose)
   BD = discrete system B matrix

6. /INPTX/ N, NX, NX1, NU, AC

   Continuous system state parameters
   AC = augmented system A matrix (codes 1 and 2)
   N = NX + NU

7. /INPTY/ NY, C /INPTW/ NW, E

   Input information for display outputs, and random inputs

8. /TIMES/ TIME, TGO, TO, TEND, DEL

   TIME = current value of time, t
   TGO = time-to-go = TEND - TIME
   TO = initial time
   TEND = final time
   DEL = discrete time step,

9. /WEIGHT/ QX, QY, QR, P

   QX, QY, QR = (augmented) state, output and control rate weightings respectively
   P = steady-state Riccati equation solution.

10. /GAINBK/ CGD

    CGD = state feedback gains

11. /INCRE/ QXI, QYI, PINC

    QXI, QYI = discrete increments to the cost functional

    PINC = increment to the steady-state Riccait equation solution

12. /STORE/ PTIME, GSAVE

    Storage for the time-varying control gains, and their corresponding times.

A5. SAMPLE PROBLEM

The sample problem for the GNPROP program is the same as that for the TIVAR program. A description of this problem is given in Section 6 of this manual. The GNPROP program is used to compute the (constant) gains for the sample problem. This section contains a listing of the user written subroutines, the input data deck, and a listing of the output.

A5.1 User Written Subroutines for the Sample Problem

```
C       GNPP - PROBLEM DEPENDENT SUBROUTINES FOR GPROPI
C       INCLUDES:
C        1 - SUBROUTINE INTOLD

        SUBROUTINE INTOLD(KEY)
C       PERFORMS INTERNAL DOWNDATES
C       PROBLEM DEPENDENT - TO BE SUPPLIED BY THE USER

        COMMON
     1  /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)

        IFLAG(KEY)=1
        RETURN

        END
```

A5.2 Input Deck for the Sample Problem

```
P-I-D CONTROLLER.  GAIN COMPUTATION
    1
    5     2     1     0     2
       0.05       -6.0       -5.0
    0     0     0     0     0     0     0     0     0     0     0     0    20    20     0     0
A
        0.0         1.0         0.0         0.0         0.0
        0.0         0.0         0.0         0.0         0.0
        0.0         0.0         0.0         1.0         0.0
        0.0         0.0       -64.0       -12.0        64.0
        0.0         0.0         0.0         0.0         0.0
B
        0.0
        0.0
        0.0
       64.0
        1.0
C
        1.0         0.0        -1.0         0.0         0.0
        0.0         1.0         0.0        -1.0         0.0
D
        0.0
        0.0
QY
        1.0         0.0
QR
   4.2206E-3
END
```

A5.3 Output Listing for the Sample Problem

```
P-I-D CONTROLLER.  GAIN COMPUTATION
 5-Dec-76    18:12

DYNAMICS READ FROM FILE: PIDTI.DYN
NO. OF TOTAL SYSTEM STATES    5
NO. OF NOISE SHAPING STATES   2
NO. OF CONTROL SYSTEM INPUTS  1
NO. OF RANDOM NOISE SOURCES   1
NO. OF DISPLAYED OUTPUTS      2

INTEGRATION TIME STEP =       0.050
INITIAL TIME  =      -6.000
TERMINAL TIME =      -5.000

INTERNAL DOWNDATES:    INDEX    CODE  NDT
                          13    PRINT   20
                          14    GNOUT   20

EXTERNAL DOWNDATE AT TIME =   -5.000    TGO =     0.000    CODE:  QY

QY    VECTOR:
    1.000E+00     0.

EXTERNAL DOWNDATE AT TIME =   -5.000    TGO =     0.000    CODE:  QR

QR    VECTOR:
    4.221E-03
DISCRETE CONTROL GAINS AT TIME =     -6.000

DGAIN MATRIX:
   -1.198E+01   -2.904E+00    4.102E+00    4.646E-01    7.878E+00
    8.840E+00

EQUIVALENT CONTINUOUS GAINS LX,LU:

CGAIN MATRIX:
   -1.540E+01   -3.335E+00    6.292E+00    6.467E-01    9.104E+00
    1.008E+01

EQUIVALENT CONTINUOUS LOPT, TN:

L*,TN MATRIX:
   -1.527E+00   -3.307E-01    6.239E-01    6.413E-02    9.027E-01
    9.916E-02
```

The following is a listing of the gains as punched by the GNPROP program. These cards are suitable for insertion directly in the TIVAR input deck.

```
TIME            -6.000
DGAIN
   -1.198E+01   -2.904E+00    4.102E+00    4.646E-01    7.878E+00
    8.840E+00
```

A6. GNPROP LISTING

```
C       NO TABS, COLONS, OR FORMFEEDS

C       GNPROP - REVERSE-TIME CONTROL GAIN PROPAGATION
C       INCLUDES
C        1 - BLOCK DATA GNDAT     - INITIALIZES VARIOUS COMMON BLOCKS
C        2 - GNMAIN               - CALLS SUBROUTINE GNPROP
C        3 - SUBROUTINE GNPROP    - PRIMARY SUBPROGRAM

C       ALSO REQUIRES THE FOLLOWING SUBPROGRAM FILES

C       GNPP - PROBLEM DEPENDENT SUBROUTINES FOR GNPROP
C       INCLUDES
C        1 - SUBROUTINE INTOLD    - PERFORMS INTERNAL DOWNDATES

C       GNPIO - I/O FOR GNPROP
C       INCLUDES
C        1 - SUBROUTINE INTLET    - SPECIFY INTERNAL DOWNDATES
C        2 - SUBROUTINE DNDATE    - PERFORMS EXTERNAL DOWNDATES
C        3 - SUBROUTINE GNRITE    - WRITES OUT THE GAINS
C
        BLOCK DATA GNDAT
C       INITIALIZES VARIOUS PARAMETERS

        COMMON
     1  /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)
     2  /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK

        DATA
     1  NCODES /16/,
     2  ICODES
     2  /1HA, 1HB, 1HC, 1HD, 1HE, 2HQX, 2HQY, 2HQU, 2HQR, 4HRICI,
     2   3HQXI, 3HQYI, 5HPRINT, 5HGNOUT, 2*5HDUMMY/,
     3  KIN, KOUT, KPTR, KPUNCH, KDISK
     3  / 5,    6,    6,      7,     8/

        END
```

```
      PROGRAM GNMAIN
     1 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
     2  PUNCH, GNFILE, TAPE7=PUNCH, TAPE8=GNFILE)
C     MAIN PROGRAM
C     CALLS SUBROUTINE GNPROP
      CALL GNPROP
      END
```

```
      SUBROUTINE GNPROP
C     PRIMARY SUBPROGRAM

      DIMENSION
     1 TITLE(8)

      COMMON
     1 /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1, COM1(15,15)
     5 /MAIN2/  COM2(15,15)
     6 /COMP1/  QD(15,15)
     7 /COMP3/  QXUD(15,4)
     8 /INPTS/  BD(15,4), ADT(15,15)
     9 /INPTX/  N, NX, NX1, NU, AC(15,15)
     A /INPTY/  NY, C(15,15)
     B /INPTW/  NW
     C /TIMES/  TIME, TGO, TO, TEND, DEL
     D /WEIGHT/ QX(30), QY(30), QR(30), P(15,15)
     E /GAINBK/ CGD(15,4)
     F /INCRE/  QXI(30), QYI(30), PINC(15,15)
     G /STORE/  PTIME(300), GSAVE(15,300)

      DATA
     1 MNIL /0/
C     SET NDIM
      NDIM=15
      NDIM1=NDIM+1

C     ZERO THE VECTORS AND MATRICES
1     DO 10 I=1,NDIM
      QX(I)=0.0
      QY(I)=0.0
      QR(I)=0.0
      QXI(I)=0.0
      QYI(I)=0.0
      DO 10 J=1,NDIM
      AC(I,J)=0.0
      C(I,J)=0.0
      P(I,J)=0.0
      PINC(I,J)=0.0
10    CONTINUE
      NGAIN=0

C     SPECIFY THE TITLE
      READ (KIN,1120) (TITLE(I), I=1,8)
1120  FORMAT (8A10)
      IF (EOF(KIN)) 400,50
50    CALL PAGEFD(KPTR,1)
      WRITE (KPTR,1125) (TITLE(I), I=1,8)
1125  FORMAT (/,1H ,8A10)
      CALL DAYTIM(KPTR)
      CALL LINEFD(KPTR,2)

C     SPECIFY THE GAINS OUTPUT MODE
      READ (KIN,1160) MODE
      WRITE (KOUT,1135) MODE
1135  FORMAT (20H GAINS OUTPUT MODE =,I5,/,1H )

C     GET THE PROBLEM DIMENSIONS
      READ (KIN,1160) NX, NX1, NU, NW, NY
1160  FORMAT (5I5)
      WRITE (KPTR,1180) NX, NX1, NU, NW, NY
1180  FORMAT (28H NO. OF TOTAL SYSTEM STATES ,I3,/,
     1 29H NO. OF NOISE SHAPING STATES ,I2,/,
```

```
     2  29H NO. OF CONTROL SYSTEM INPUTS,I2,/,
     3  29H NO. OF RANDOM NOISE SOURCES ,I2,/,
     4  29H NO. OF DISPLAYED OUTPUTS    ,I2,/,
     5  1H )
C       SPECIFY DEL, TO, AND TEND
        READ (KIN,1200) DEL, TO, TEND
1200    FORMAT (3E10.0)
        TEND=TO+IFIX((TEND-TO+0.001)/DEL)*DEL
        WRITE (KPTR,1225) DEL, TO, TEND
1225    FORMAT (25H INTEGRATION TIME STEP = ,F10.3,/,
     1  17H INITIAL TIME  = ,F10.3,/,
     2  17H TERMINAL TIME = ,F10.3,/,
     3  1H )
C       SPECIFY INTERNAL DOWNDATES
        CALL INTLET
        WRITE (KPTR,1310)
1310    FORMAT (22H INTERNAL DOWNDATES   ,19H INDEX    CODE  NDT)
        DO 65 I=1,NCODES
        IF (NSTEP(I) .EQ. 0) GO TO 65
        WRITE (KPTR,1320) I, ICODES(I), NSTEP(I)
1320    FORMAT (26X, I2, 3X, A5, 3X, I2)
        IF (MOD(INT((TEND-TO+0.001)/DEL),NSTEP(I)) .EQ. 0) GO TO 65
        WRITE (KPTR,1330)
1330    FORMAT (1H ,16H   ***WARNING***,/,
     1  1H ,55H(TF-TO) HAS A NON-INTEGRAL NUMBER OF INTERNAL DOWNDATES)
65      CONTINUE
C       INITIALIZE SOME MORE QUANTITIES
        NTERMS=5
        NXP1=NX+1
        N=NX+NU
        TIME=TEND
        TGO=TEND-TIME
        TGNEXT=0.0
C       MAIN COMPUTATIONAL LOOP STARTS HERE
C       START BY HANDLING INTERNAL AND EXTERNAL DOWNDATES
70      CALL DNDATE(TGNEXT)
C       SKIP THIS PART IF NO PRINTING WAS CALLED FOR, OR IF TGO IS 0
80      IF (IFLAG(13) .EQ. 0) GO TO 100
        IF (TGO .LE. 0.0) GO TO 100
        CALL TRANS2(N,NU,CGD,COM2)
        WRITE (KPTR,2090) TIME
2090    FORMAT (1H ,33HDISCRETE CONTROL GAINS AT TIME = ,F10.3)
        CALL MATIO(COM2,NU,N,3)
        WRITE (KPTR,2092)
2092    FORMAT (1H ,34HEQUIVALENT CONTINUOUS GAINS LX,LU )
        DO 95 L=1,2
        CALL EQUATE(QD,AC,N,N)
        CALL MSCALE(QD(NXP1,1),COM2,NU,N,-1.0)
        CALL DSCRT(N,QD,DEL,COM2,COM1,4)
        CALL GMINV(N,N,COM1,QD,MR,1)
        CALL MAT5A(CGD,QD,NU,N,N,COM2)
        CALL MSCALE(COM2,COM2,NU,N,DEL)
95      CONTINUE
        CALL MATIO(COM2,NU,N,3)
        CALL GMINV(NU,NU,COM2(1,NXP1),COM1(1,NXP1),MR,1)
        CALL MMUL(COM1(1,NXP1),COM2,NU,NU,NX,COM1)
        WRITE (KPTR,2096)
2096    FORMAT (1H ,31HEQUIVALENT CONTINUOUS LOPT, TN )
        CALL MATIO(COM1,NU,N,3)
```

```
C       SAVE THE GAINS
100     IF (IFLAG(14).EQ.0 .AND. TIME.GT.(TO+0.001)) GO TO 110
        IF (TIME-0.001 .LE. TO) TIME=TO
        IF (MODE .EQ. 0) GO TO 110
        IF (TGO .LE. 0.0) GO TO 110
        L=NGAIN*NU+1
        NGAIN=NGAIN+1
        PTIME(NGAIN)=TIME
        CALL EQUATE(GSAVE(1,L),CGD,N,NU)
        IF (L .LT. 300) GO TO 110
        WRITE (KPTR,2105)
2105    FORMAT (1H ,37HINCREASE STORAGE ALLOCATION FOR GAINS)
        CALL EXIT

C       PROPAGATE THE GAINS FOR ANOTHER TIME STEP
110     IF (TIME .EQ. TO) GO TO 300
        IF (IFLAG(10) .EQ. 1) CALL MADD1(N,N,P,PINC,P,1.0)
        IF (IFLAG(11) .EQ. 0) GO TO 172
        DO 171 I=1,N
        P(I,I)=P(I,I)+QX(I)
171     CONTINUE
172     IF (IFLAG(12) .EQ. 0) GO TO 174
        DO 173 I=1,NY
        C1=QYI(I)
        DO 173 J=1,N
        COM1(J,I)=C1*C(I,J)
173     CONTINUE
        CALL MMULS(COM1,C,N,NY,N,P)

C       DISCRETIZE NEW SYSTEM DYNAMICS
C       ADT, BD ARE DISCRETIZED AT B OVER THE INTERVAL (TIME-DEL,TIME)
174     IF (IFLAG(1)+IFLAG(2) .EQ. 0) GO TO 175
        CALL DSCRT(N,AC,DEL,ADT,COM2,NTERMS)
        CALL TRANS1(N,ADT,ADT)
        CALL EQUATE(BD,COM2(1,NXP1),N,NU)

C       OBTAIN DISCRETE COST WEIGHTS
175     DO 177 I=1,NY
        C1=QY(I)*DEL/2.0
        DO 177 J=1,N
        QD(I,J)=C1*C(I,J)
177     CONTINUE
        CALL MAT2A(NY,N,C,QD,QD)
        DO 178 I=1,N
        QD(I,I)=QD(I,I)+QX(I)*DEL/2.0
178     CONTINUE
        CALL MMUL(ADT,QD,N,N,N,COM1)
        DO 179 I=1,NU
        C1=-1.0/(QR(I)*DEL)
        DO 179 J=1,N
        COM2(J,I)=QD(J,I+NX)*DEL
        QXUD(J,I)=C1*COM2(J,I)
179     CONTINUE
        CALL MAT6S(N,N,COM1,ADT,QD)
        CALL MAT6S(N,NU,COM2,QXUD,QD)

C       UPDATE CYCLE
        CALL MMUL(P,BD,N,N,NU,COM1)
        CALL MAT2A(N,NU,COM1,BD,CGD)
        DO 180 I=1,NU
        CGD(I,I)=CGD(I,I)+QR(I)*DEL
180     CONTINUE
        CALL GMINV(NU,NU,CGD,COM2,MR,1)
        CALL MMUL(COM1,COM2,N,NU,NU,CGD)
        CALL MAT5(CGD,BD,N,NU,N,COM1)
        DO 183 I=1,N
```

```
      DO 182 J=1,N
      COM1(I,J)=-COM1(I,J)
182   CONTINUE
      COM1(I,I)=COM1(I,I)+1.0
183   CONTINUE
      CALL MMUL(COM1,P,N,N,N,COM2)
      DO 185 I=1,NU
      C1=QR(I)*DEL
      DO 185 J=1,N
      P(J,I)=C1*CGD(J,I)
185   CONTINUE
      CALL MAT2(N,NU,CGD,P,P)
      CALL MAT6S(N,N,COM2,COM1,P)

C     PROPAGATE CYCLE
      TIME=TIME-DEL
      TGO=TGO+DEL
      CALL EQUATE(COM1,ADT,N,N)
      CALL MAT5S(QXUD,BD,N,NU,N,COM1)
      CALL MMUL(COM1,P,N,N,N,COM2)
      CALL MAT6S(N,N,COM2,COM1,QD)
      CALL EQUATE(P,QD,N,N)

C     GET NEW GAINS
      CALL MSCALE(QXUD,QXUD,N,NU,-1.0)
      CALL MMULS(COM1,CGD,N,N,NU,QXUD)
      CALL EQUATE(CGD,QXUD,N,NU)
      GO TO 70

C     TIME IS EXPIRED - WRITE OUT THE GAINS ON A FILE
300   N1=NU
      N2=N
      CALL GNRITE(MODE,NGAIN,N1,N2)

C     DONE - RETURN
400   RETURN
      END
```

```
C       GNPIO - I/O FOR GNPROP
C       INCLUDES
C        1 - SUBROUTINE INTLET
C        2 - SUBROUTINE DNDATE
C        3 - SUBROUTINE GNRITE


        SUBROUTINE INTLET
C       SPECIFY INTERNAL DOWNDATES
C       NSTEP(I) REFERS TO THE ITH TYPE OF DOWNDATE
C       NSTEP(I)=0  NO INTERNAL DOWNDATE
C       NSTEP(I)=NDT  NDT DELS BETWEEN SUCCESSIVE INTERNAL DOWNDATES

        COMMON
     1  /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)
     2  /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK


C       READ NSTEP(I) FROM ONE CARD
100     READ (KIN,1100) (NSTEP(I), I=1,NCODES)
1100    FORMAT (16I5)
        RETURN
        END
```

```
      SUBROUTINE DNDATE(TGNEXT)
C     PERFORMS EXTERNAL DOWNDATES

      COMMON
     1 /COMMUN/ NCODES, ICODES(16), IFLAG(16), NSTEP(16)
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1, COM1(1)
     9 /INPTX/  N, NX, NX1, NU, AC(1)
     A /INPTY/  NY, C(1)
     B /INPTW/  NW
     C /TIMES/  TIME, TGO, TO, TEND, DEL
     D /WEIGHT/ QX(30), QY(30), QR(30)
     F /INCRE/  QXI(30), QYI(30), PINC(1)

      DATA
     1 LEND, LTIME, LTGO /3HEND, 4HTIME, 3HTGO/

      NNX=NX*NDIM+1
      NXP1=NX+1

C     TAKE CARE OF INTERNAL DOWNDATES
      DO 100 I=1,NCODES
      IFLAG(I)=0
      IF (NSTEP(I).EQ.0) GO TO 100
      IF (MOD(INT(TGO/DEL+0.01),NSTEP(I)) .EQ. 0) CALL INTOLD(I)
100   CONTINUE

C     RETURN IF NOT TIME FOR THE NEXT EXTERNAL DOWNDATE
110   IF (TGO+0.001  .LT. TGNEXT) GO TO 220
      WRITE (KPTR,1010) TGO
1010  FORMAT (/,1H ,23HEXT. DOWNDATE AT TGO = ,F10.3)

C     SPECIFY THE TYPE OF EXTERNAL DOWNDATE (AND TGNEXT IF LTGO)
120   READ (KIN,1030) IDEN, BRKT
1030  FORMAT (A5,5X,E10.0)
      IF (EOF(KIN)) 135,130

C     CHECK FOR LEND
130   IF (IDEN.NE.LEND) GO TO 140
135   TGNEXT=TEND-TO+0.1
      GO TO 110

C     CHECK FOR LTGO
140   IF (IDEN.NE.LTGO) GO TO 145
      TGNEXT=BRKT
      GO TO 110

C     CHECK FOR LTIME
145   IF (IDEN.NE.LTIME) GO TO 150
      TGNEXT=TEND-BRKT
      GO TO 110

C     DONE WITH SPECIAL CHECKS
150   CONTINUE

C     SEARCH THROUGH THE DOWNDATE CODES, ICODE(KEY)
      DO 160 KEY=1,NCODES
      IF (IDEN.EQ.ICODES(KEY)) GO TO 170
160   CONTINUE

C     CODE WAS ILLEGAL
      WRITE (KPTR,1165) IDEN
1165  FORMAT (1H ,5HCODE ,A5,11H IS ILLEGAL)
      CALL EXIT
```

```
C       DO THE SPECIFIED EXTERNAL DOWNDATE
170     IFLAG(KEY)=1
        IO=2
        WRITE (KPTR,1175) TIME, TGO, IDEN
1175    FORMAT (/,29H EXTERNAL DOWNDATE AT TIME = ,F8.3,
     1  4X,6HTGO = ,F8.3,4X,5HCODE ,2X,A5)
        GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16), KEY

C       SYSTEM DYNAMICS - A, B, C, D, E
1       CALL MATIO(AC,NX,NX,IO)
        GO TO 120
2       CALL MATIO(AC(NNX),NX,NU,IO)
        GO TO 120
3       CALL MATIO(C,NY,NX,IO)
        GO TO 120
4       CALL MATIO(C(NNX),NY,NU,IO)
        GO TO 120
5       CALL MATIO(COM1,NX,NW,IO)
        GO TO 120

C       COST FUNCTIONAL - QX, QY, QU, QR
6       CALL VECTIO(QX,NX,IO)
        GO TO 120
7       CALL VECTIO(QY,NY,IO)
        GO TO 120
8       CALL VECTIO(QX(NXP1),NU,IO)
        GO TO 120
9       CALL VECTIO(QR,NU,IO)
        GO TO 120

C       RICCATI INCREMENT - RICI
10      CALL MATIO(PINC,N,N,IO)
        GO TO 120

C       COST INCREMENTS - QXI, QYI
11      CALL VECTIO(QXI,NX,IO)
        GO TO 120
12      CALL VECTIO(QYI,NY,IO)
        GO TO 120

C       PRINT THE GAINS - PRINT
13      CONTINUE
        GO TO 120

C       SAVE THE GAINS - GNOUT
14      CONTINUE
        GO TO 120

C       DUMMY DOWNDATES - DUMMY
15      CONTINUE
16      CONTINUE
        GO TO 120

C       NO MORE EXTERNAL DOWNDATES AT THIS TIME
220     CONTINUE
        RETURN

        END
```

```
      SUBROUTINE GNRITE(MODE,NGN,NU,N)
C     WRITE OUT THE (TRANSPOSED) GAINS IN REVERSE ORDER FOR TIVAR
C     MODE=1  PUNCH
C     MODE=2  WRITE ON A FILE

      COMMON
     2 /INOU/   KIN, KOUT, KPTR, KPUNCH, KDISK
     4 /MAIN1/  NDIM, NDIM1, COM1(1)
     G /STORE/  PTIME(300), GSAVE(1)

      DATA
     1 LTIME, LDGAIN /4HTIME, 5HDGAIN/

      K=NGN+1
      L=NGN*NU*NDIM+1
      JEND=N*NDIM

C     INITIALIZE THE FILE UNIT NUMBER - KFIL
10    IF (MODE.LE.0) RETURN
      IF (MODE.EQ.1) KFIL=KPUNCH
      IF (MODE.EQ.2) KFIL=KDISK
      IF (MODE.GE.3) RETURN

C     OUTPUT THE GAINS
100   L=L-NU*NDIM
      K=K-1
      IF (K.EQ.0) GO TO 200
      TIME=PTIME(K)
      CALL TRANS2(N,NU,GSAVE(L),COM1)
      WRITE (KFIL,1020) LTIME, TIME, LDGAIN
1020  FORMAT (A4,6X,F10.3,/,A5)
      DO 120 I=1,NU
      WRITE (KFIL,1040) (COM1(IJ), IJ=I,JEND,NDIM)
1040  FORMAT (1P8E10.3)
120   CONTINUE
      GO TO 100

C     DONE
200   RETURN

      END
```

C LIBRARY FOR OPTIMAL CONTROL MODEL

The following subroutines from the Optimal Control Model Library are used in program GNPROP, as well as in TIVAR. Listing for these subprograms may be found in Section 7 of this manual.

```
SUBROUTINE DSCRT(N,A,DEL,EA,EAINT,NT)
SUBROUTINE GMINV(NR,NC,A,U,MR,MT)
SUBROUTINE MAT2(N1,N2,X,Y,Z)
SUBROUTINE MAT2A(N1,N2,X,Y,Z)
SUBROUTINE MAT5(X,Y,N1,N2,N3,Z)
SUBROUTINE MAT5A(X,Y,N1,N2,N3,Z)
SUBROUTINE MAT6(N1,N2,X,Y,Z)
SUBROUTINE MMUL(X,Y,N1,N2,N3,Z)
SUBROUTINE MADD1(NR,NC,X,Y,Z,C1)
SUBROUTINE EQUATE(A,B,NR,NC)
SUBROUTINE MSCALE(A,B,NR,NC,C1)
SUBROUTINE TRANS1(N,A,AT)
SUBROUTINE TRANSP(N1,N2,X,XPOSE)
FUNCTION DOT(NR,A,B)
FUNCTION DOT2(NN,A,B)
SUBROUTINE VADD(N,C1,A,B)
SUBROUTINE VADD1(NN,C1,A,B)
SUBROUTINE VSCALE(X,Y,N,C1)
SUBROUTINE OUTPUT(M,N,X,LA)
SUBROUTINE VECOUT(X,N,LET)
SUBROUTINE MATIO(X,NR,NC,IO)
SUBROUTINE VECTIO(X,N,IO)
SUBROUTINE KVECIO(IX,N,IO)
SUBROUTINE PAGEFD(KFIL,KOUNT)
SUBROUTINE LINEFD(KFIL,KOUNT)
SUBROUTINE DAYTIM(KFIL)
```

DISTRIBUTION

	No. of Copies
Assistant Secretary of Defense for Manpower and Reserve Affairs Department of Defense Washington, D.C. 20301	1
HQ DA DAPE-PBR, Mr. Barber Washington, D.C. 20310	1
Commander US Army Materiel Development and Readiness Command Attn: DRXAM-TL 5001 Eisenhower Avenue Alexandria, Virginia 22333	2
Headquarters US Army Medical R&D Command Attn: Behavior Sciences Rsch Br. Main Navy Building Washington, D.C. 20315	1
Director Walter Reed Army Institute of Research Walter Reed Army Medical Center Attn: Neuropsychiatry Division Washington, D.C. 20012	1
OAD/E&LS ODDR&E, Pentagon, Room 3D129 Attn: LTC Henry L. Taylor Washington, D.C. 20310	1
US Army Human Engineering Laboratory Representative Attn: DRXHE-AV, Dr. M. Hofmann Directorate for RD&E US Army Aviation Systems Command P. O. Box 209 St. Louis, Missouri 63166	1
Commander US Naval Weapons Laboratory Attn: Technical Library Dahlgren, Virginia 22448	1
Commander US Naval Weapons Center Attn: Code 533, Technical Library China Lake, California 93555	1
Director, Development Center US Marine Corps Development and Education Command Attn: Chairman, Air Operations Division Quantico, California 22134	1
HQ Rome Air Development Center Attn: TILD Griffiss Air Force Base New York 13440	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-AM, Mr. Miles US Army Systems Analysis Activity Aberdeen Proving Ground Maryland 21005	1

	No of Copies
Chief Human Engineering Laboratory Detachment Attn: DRXHE-FA, Mr. Sorin Frankford Arsenal Philadelphia Pennsylvania 19137	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-PTA, Mr. Carlock Picatinny Arsenal Dover, New Jersey 07801	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-RI, Mr. Galbauy Rock Island Arsenal Rock Island, Illinois 61201	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-TA, Mr. J. Erickson US Army Tank-Automotive Research and Development Command Warren, Mississippi 48090	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-ME, Mr. Antenucci, c/o DRXFB-TO US Army Mobility Equipment Research and Development Command Fort Belvoir, Virginia 22060	1
US Army Research Office Attn: Information Processing Office Box CM, Duke Station Durham, North Carolina 27706	1
Commander US Army Air Defense School Attn: ATSA-CTD-MO Fort Bliss, Texas 79916	1
Commander Rock Island Arsenal Attn: SARRI-LP-L, Technical Library Rock Island, Illinois 61201	1
Commander US Army Electronics Command Attn: DRSEL-PA-RH, Mr. Baron Fort Monmouth, New Jersey 07703	1
Library US Army War College Carlisle Barracks Pennsylvania 17013	1
Director US Army Ballistic Research Laboratory Attn: DRXBR-EB Aberdeen Proving Ground Maryland 21005	1

Address	No. of Copies
Army Research Institute Field Unit Attn: Dr. Robert W. Bauer Building 2423 Fort Knox, Kentucky 40121	1
Department of Transportation Library Reference and Research Branch, TAD-494.6 800 Independence Avenue SW Washington, D.C. 20591	1
US Postal Service Laboratory Attn: Mr. D.Y. Cornog Chief, HF Group 11711 Parklawn Drive Rockville, Maryland 20852	1
Mr. Edgar M. Johnson US Army Research Institute Room 239, The Commonwealth Building 1320 Wilson Boulevard Arlington, Virginia 22209	1
Federal Aviation Administration Civil Aeromedical Institute Library AC-101.1, P. O. Box 25082 Oklahoma City, Oklahoma 73125	1
Department of Operations Analysis Naval Postgraduate School Attn: Professor James K. Arima Monterey, California 93940	1
Commander USAVSCOM Attn: DRSAV-R-F, Mr. S. Moreland P. O. Box 209 St. Louis, Missouri 63166	1
Director Naval Research Laboratory Attn: Code 5132A Washington, D.C. 20390	1
Commanding Officer Naval Training Equipment Center Attn: Technical Library Orlando, Florida 32813	1
6570 AMRL Attn: MRHE MRHE, Dr. M. J. Warrick MRHER, Mr. C. Bates, Jr. Wright-Patterson Air Force Base Ohio 45433	 2 1 1
AF Flight Dynamics Laboratory Attn: FDCR (CDIC) AFFDC-FGR, LT Greg Peters Wright-Patterson Air Force Base Ohio 45433	 1 1
Director US Army Human Engineering Laboratory Attn: DRXHE-DBD Aberdeen Proving Ground Maryland 21005	2
Medical Library, Building 148 Naval Submarine Medical Research Laboratory Box 900, Submarine Base New London Groton, Connecticut 06340	1
Code 455 Office of Naval Research Washington, D.C. 20360	1
Dr. Marshall J. Farr Associate Director, Personnel and Training Code 458 Office of Naval Research Washington, D.C. 20360	1
Commandant US Army Artillery and Missile School Attn: USAAMS Technical Library Fort Sill, Oklahoma 73503	1
ARI Field Unit P. O. Box 6057 Fort Bliss, Texas 79916	1
Commander Fort Huachuca Support Command, US Army Attn: Tech References Division Fort Huachuca, Arizona 85613	1
Commander Attn: Technical Library White Sands Missile Range New Mexico 88002	1
Director USA Air Mobility Research and Development Laboratory Attn: Dr. Richard S. Dunn Ames Research Center Moffett Field, California 94035	1
Commander US Army Electronics Command Attn: DRSEL-VL-E Fort Monmouth, New Jersey 07703	1
Director Military Psychology and Leadership United States Military Academy West Point, New York 10996	1
Commander Watervliet Arsenal Attn: SWEWV-RDT Watervliet, New York 12189	1
Commander Frankford Arsenal Attn: Library (C2500, Building 51-2) Philadelphia, Pennsylvania 19137	1

	No. of Copies
ARI Field Unit Attn: Library Fort Knox, Kentucky 40121	1
Commander US Army Tank-Automotive Command Attn: DRSTA-RHFL, Research Library Warren, Mississippi 48090	1
Commander US Army Tank-Automotive Command Attn: DRSTA-R Warren, Mississippi 48090	1
Director of Graduate Studies and Research Attn: Behavioral Sciences Representative US Army Command and General Staff College Fort Leavenworth, Kansas 66027	1
US Army Agency for Aviation Safety Attn: Librarian Fort Rucker, Alabama 36360	1
Chief Army Research Institute P. O. Box 476 Fort Rucker, Alabama 36360	1
Commander US Army Aeromedical Research Laboratory P. O. Box 577 Attn: Dr. Kent Kimball	1
Library	1
Fort Rucker, Alabama 36360	
US Army Natick Research and Development Command Attn: Tech Library (DRXNM-STL) Natick, Massachusetts 01760	1
US Army Natick Research and Development Command Behavioral Sciences Division Attn: DRXNM-PRB	1
DRXNM-PRBE	1
Natick, Massachusetts 01760	
US Army Medical Bioengineering Research and Development Laboratory Fort Detrick, Building 568 Frederick, Maryland 21701	1
Commander US Army Training Device Agency, Naval Training Equipment Center Attn: Code N2A Orlando, Florida 32813	1

	No. of Copies
US Army Infantry School Library Infantry Hall Fort Benning, Georgia 31905	1
Project Manager Office of the Project Manager for Training Devices Attn: DRCPM-TND, Dr. R.E. Odom Fort Benning, Georgia 31905	1
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Commander US Army Materiel and Readiness Command Attn: DRCRD	1
DRCDL	1
5001 Eisenhower Avenue Alexandria, Virginia 22333	
Bolt, Beranek and Newman Inc. Attn: Dr. S. Baron	2
Dr. D. Kleinman	1
Dr. J. Baliser	1
50 Moulton Street Cambridge, Maryland 02138	
DRXHE-MI, Mr. Chaikin	1
DRCPM-HFE	1
-AH	1
-MPE	1
-LDE	1
-ROL-E	1
-RK	1
-HEL-T	1
-TOE	1
-MW	1
DRSMI-LP, Mr. Voigt	1
-W	1
DRDMI-X, Dr. McDaniel	1
-T, Dr. Kobler	1
-H	1
-Y	1
-YD	1
-YT	1
-TD	1
-TDC	1
-TDD	1
-TDF	1
-TDK	1
-TDR	1
-TDS	1
-TDW	1
-TDW, Mr. Dickson	5
-TE	1
-TG	1
-TL	1
-TPN, L. Aymett	1
-TBD	3
-TI (Record Set)	1
(Reference Copy)	1